

DUDLEY KNOX LIBRARY
NAVAL POSTGRADUATE SCHOOL
MONTEREY, CALIFORNIA 93943-8002

NAVAL POSTGRADUATE SCHOOL

Monterey, California



THESIS

AUTOMATED POLE PLACEMENT ALGORITHM

FOR MULTIVARIABLE OPTIMAL CONTROL SYNTHESIS

by

Chow, Wah Keh

September 1985

Thesis Advisor:

D. J. Collins

Approved for public release; distribution unlimited.

T226057

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER	2. GOVT ACCESSION NO.	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) Automated Pole Placement Algorithm for Multivariable Optimal Control Synthesis		5. TYPE OF REPORT & PERIOD COVERED Master's Thesis September 1985
		6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s) Chow, Wah Keh		8. CONTRACT OR GRANT NUMBER(s)
9. PERFORMING ORGANIZATION NAME AND ADDRESS Naval Postgraduate School Monterey, California 93943		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS
11. CONTROLLING OFFICE NAME AND ADDRESS Naval Postgraduate School Monterey, California 93943		12. REPORT DATE September 1985
		13. NUMBER OF PAGES 170
14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office)		15. SECURITY CLASS. (of this report) Unclassified
		15a. DECLASSIFICATION/DOWNGRADING SCHEDULE
16. DISTRIBUTION STATEMENT (of this Report) Approved for public release; distribution unlimited.		
17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)		
18. SUPPLEMENTARY NOTES		
19. KEY WORDS (Continue on reverse side if necessary and identify by block number) Linear Quadratic Control, Pole Placement, Pole Assignment, Multivariable System, Multi-input Multi-output Control, Multivariable System, Robustness, Weighting Matrix.		
20. ABSTRACT (Continue on reverse side if necessary and identify by block number) This work addresses the application of numerical optimization technique to the pole-placement problem in multivariable optimal control. An algorithm is developed to select a set of weighting matrix element such that the conventional transient response criteria are satisfied. General properties of the optimal system in terms of stability, robustness and relative weights between state and control		

variables were explored by applying the method to the design of two multivariable systems. Results indicated that this method provides good insight to the problem for the designer and is therefore a useful tool in multivariable control synthesis.

Approved for public release; distribution is unlimited.

Automated Pole Placement Algorithm
for Multivariable Optimal Control Synthesis

by

Chow, Wah Keh
B.A.(Hons), University of Oxford, 1978

Submitted in partial fulfillment of the
requirements for the degree of

MASTER OF SCIENCE IN AERONAUTICAL ENGINEERING

from the

NAVAL POSTGRADUATE SCHOOL
September 1985

Thesis
2008
11

ABSTRACT

This work addresses the application of numerical optimization technique to the pole-placement problem in multivariable optimal control. An algorithm is developed to select a set of weighting matrix element such that the conventional transient response criteria are satisfied.

General properties of the optimal system in terms of stability, robustness and relative weights between state and control variables were explored by applying the method to the design of two multivariable systems. Results indicated that this method provides good insight to the problem for the designer and is therefore a useful tool in multivariable control synthesis.

TABLE OF CONTENTS

I.	INTRODUCTION	11
A.	BACKGROUND	11
B.	OVERVIEW	13
II.	MULTIVARIABLE LINEAR QUADRATIC CONTROL	14
A.	LINEAR QUADRATIC SYSTEM	14
B.	POLE ASSIGNMENT PROBLEM	17
C.	WEIGHTING MATRICES AND SYSTEM PERFORMANCES . .	19
III.	ROBUSTNESS THEORY AND LINEAR QUADRATIC CONTROL . .	24
A.	ROBUSTNESS CONCEPTS	24
B.	ROBUSTNESS IN LQ SYSTEM	30
IV.	DESIGN PROCEDURE	32
A.	GENERAL	32
B.	APPROACHES	32
C.	POLE PLACEMENT ALGORITHMS	34
	1. System with Real and Distinct Eigenvalues	34
	2. System with Complex Eigenvalues	37
	3. System With Repeated Eigenvalues	38
D.	COMPUTER PROGRAM DESCRIPTION	39
E.	DESIGN PHILOSOPHY	40
V.	DESIGN EXAMPLES	44
A.	INTRODUCTORY 2X2 PROBLEM	44
B.	A HELICOPTER DESIGN PROBLEM	56
C.	DISCUSSION AND CONCLUDING REMARKS	82
VI.	CONCLUSIONS	94

APPENDIX A: NON-DIAGONAL R DESIGNS	96
APPENDIX B: EXAMPLE DESIGN RUN OUTPUTS	106
APPENDIX C: COMPUTER PROGRAM LISTINGS	116
LIST OF REFERENCES	168
INITIAL DISTRIBUTION LIST	170

LIST OF TABLES

I	PERTURBED EIGENVALUE FOR 2X2 MODEL	48
II	PERTURBED EIGENVALUE FOR LQ DESIGN	53
III	ALPHATECH DESIGNS (1, 2, AND 3)	65
IV	RESULTS FROM POLE PLACEMENT SEQUENCE(LQ-A) . . .	67
V	RESULTS FROM POLE PLACEMENT SEQUENCE(LQ-B) . . .	68
VI	CLOSED-LOOP EIGENVECTORS	69
VII	RESULTS FROM POLE PLACEMENT SEQUENCE (LQ-C) . . .	99
VIII	RESULTS FROM POLE PLACEMENT SEQUENCE (LQ-D) . .	100

LIST OF FIGURES

2.1	State Feedback System-Time and Frequency Domain . .	16
2.2	Selecting Relative Weighting Parameter ρ	20
2.3	Single Input Asymptotic Root Loci	22
2.4	Multiple Input Asymptotic Root Loci	23
3.1	The Perturbed SISO System	25
3.2	The Nyquist Plot - Gain and Phase Margin	26
3.3	The Minimum Guaranteed GM and PM	27
3.4	The Perturbed MIMO System	28
3.5	SISO LQ Nyquist	30
4.1	Computer Aided Design Package Organization	41
4.2	LQ Eigenvalue Assignment Philosophy	43
5.1	A Simple 2X2 MIMO Model	45
5.2	Unity Feedback for the 2X2 Model	47
5.3	Loop Transfer Matrix - $F = I$	49
5.4	Multivariable Nyquist Plot	50
5.5	Singular Value Plots - Unity Feedback	51
5.6	Comparison-Singular Value Plots (2x2 model)	54
5.7	Closed-Loop Pole-Zero Plots (Unity Gain FB)	56
5.8	Closed-Loop Pole-Zero Plots (LQ Design)	57
5.9	Open-Loop Bode Plots Channel 1-1	58
5.10	Open-Loop Bode Plots Channel 2-1	59
5.11	Open-Loop Bode Plots Channel 2-2	60
5.12	State Variables and Body Axes for CH-47	62
5.13	Open Loop Time Response $\phi(0) = 0.1$ rad	64
5.14	Closed-Loop Time Response Plot (AlphaTech 1)	70
5.15	Closed-Loop Time Response Plot (LQ-A)	71
5.16	Closed-Loop Time Response Plot (LQ-B)	72
5.17	Closed-Loop Pole-Zero Plots (AlphaTech 1)	74

5.18	Closed-Loop Pole-Zero Plots (LQ-A)	75
5.19	Closed-Loop Pole-Zero Plots (LQ-B)	76
5.20	Bode Plots Comparison- Input 1-1	78
5.21	Bode Plots Comparison - Input 1-2	79
5.22	Bode Plots Comparison - Input 2-1	80
5.23	Bode Plots Comparison - Input 2-2	81
5.24	Closed-Loop Bode Plot- AlphaTech 1-1	83
5.25	Closed-Loop Bode Plot- AlphaTech 1-3	84
5.26	Closed-Loop Bode Plot- AlphaTech 1-4	85
5.27	Closed-Loop Bode Plot- LQ-A 1-1	86
5.28	Closed-Loop Bode Plot- LQ-A 1-3	87
5.29	Closed-Loop Bode Plot- LQ-A 1-4	88
5.30	Closed-Loop Bode Plot- LQ-B 1-1	89
5.31	Closed-Loop Bode Plot- LQ-B 1-3	90
5.32	Closed-Loop Bode Plot- LQ-B 1-4	91
A.1	Singular Value Plots - Comparison	101
A.2	Bode Plots Comparison- Input 1-1	102
A.3	Bode Plots Comparison - Input 1-2	103
A.4	Bode Plots Comparison - Input 2-1	104
A.5	Bode Plots Comparison - Input 2-2	105

ACKNOWLEDGMENTS

I wish to gratefully acknowledge my thesis advisor, Prof. D. J. Collins, for suggesting the basis of this thesis work, and for his invaluable advice and guidance during the course of the thesis.

I would also like to express my appreciation to Prof. J. V. Sanders and R. D. Zucker for their support during my study at NPS, and to all others who have helped me directly or indirectly to successfully complete this thesis.

I. INTRODUCTION

A. BACKGROUND

For more than two decades, the need to solve control problems in aerospace application has been the primary driving force behind the modern control theory development. Problems in manoeuvring, guidance and tracking of aircrafts and space vehicles have motivated the development of various control design and synthesis methods. One of the methods, the so-called Linear Quadratic Control or LQ-control [Refs. 1,2] has been widely used and is treated extensively in the control literatures. Unlike most classical methods where the design are based on conventional time response criteria, the LQ theory treats the problem of designing controllers as that of minimizing a quadratic cost function of states and control inputs. The design problems become that of selecting suitable weighting matrices in the performance index. Two questions naturally arise from this method. First, how does one select the weighting matrices and second, once a set weighting matrices is selected, how does one know that it is a good design. There are generally two approaches to the first problem; the obvious one is to rely on physical arguments and a certain amount of trial and error [Refs. 3,4] Unfortunately, such formulation can be obtained in only a few cases. Reference 4 provides a few guidelines that can be employed. The second approach is to avoid the physical aspect of the performance index but instead try to relate the weighting matrices with some other performance specification. For example, Tyler and Tuteur [Ref. 5] expressed the characteristic polynomial as an explicit function of the weighting matrices for single input

single output (SISO) system with diagonal weighting elements. Root-Locus type of procedure were used to show how the variation in weighting elements affect the eigenvalues of the closed-loop system. Similar relations were explored in [Refs. 6,7] in which more general expressions were obtained. Their uses were restricted to single input case due to difficulty in handling polynomial matrices. Solheim [Ref. 8] later developed a sequential design procedure based on diagonalized (decoupled) system. More recent results of eigenvalue placement in optimal control problem are presented in [Refs. 9,10,11,12].

The second issue of LQ design, i.e., whether a good design has been obtained once the performance index has been fixed, is related to the multivariable nature of system. Unlike the single input case where the closed-loop eigenvalues uniquely define the feedback gain and hence the weighting matrices, the MIMO structure provides additional dimension which allows further tradeoff for properties other than the closed-loop eigenvalue location. An example is the gain and phase margin, or in MIMO case, the so-called robustness criteria. Robust-control has been the subject of extensive researches in recent years and results relating to optimal control can be found in [Refs. 13,14,15].

It becomes evident from the above discussion that LQ control design and synthesis are not just a matter of specifying performance index. Optimal in the sense of satisfying performance index and perhaps pole location does not necessarily means that a good design has been obtained. Other criteria like disturbance rejection, robustness and sensitivity need to be considered and incorporated in the design procedure. This is the motivation behind our present research which in turn leads to the development of the synthesis package. An overview of the thesis is described in the next section.

B. OVERVIEW

In this section an overview of the thesis is given. A background of multivariable optimal control theory in terms of its structure, frequency domain characteristic and asymptotic properties is first discussed together with derivation of some useful relationships in Chapter 2. General robustness concepts and its application in Linear Quadratic (LQ) Control is presented in Chapter 3. In Chapter 4, a computer aided design package for pole-placement synthesis based on numerical optimization technique is presented. This package provides a useful computational tool to support the material in the remaining chapter. A step-by-step pole-placement synthesis procedure is also presented to illustrate how the package can be used to design optimal control system that meet time response criteria and other properties. The use of the pole-placement synthesis package in two actual design problems is demonstrated in Chapter 5. Results in terms of frequency and time response properties, robustness (singular value decomposition) are compared to those obtained by other design methods. Program listings and an example of a design session are included in the Appendix.

II. MULTIVARIABLE LINEAR QUADRATIC CONTROL

In this Chapter a brief review of Multivariable Linear Quadratic Control theory is presented. Structure of LQ system is first given, general stability properties are then discussed. In section two the general pole assignment problem is formulated for the MIMO state feedback system. Interpretation of the state and control input weighting matrices and their effect on the closed-loop time response behaviors and the asymptotic properties are presented.

A. LINEAR QUADRATIC SYSTEM

Consider the following linear time-invariant state-space system given by the equations:

$$\dot{x}(t) = Ax(t) + Bu(t) \quad (\text{eqn 2.1})$$

$$y(t) = Cx(t) \quad (\text{eqn 2.2})$$

where $x(t)$ is the n -dimensional state vector, $u(t)$ is the m -dimensional control vector and $y(t)$ is the p -dimensional output vector. A , B and C are real constant matrices of dimension $n \times n$, $n \times m$ and $p \times n$ respectively. Assuming that they form a controllable pair (A, B) and an observable pair (A, C) , the optimal feedback control law is obtained by minimizing the following quadratic performance index.

$$J = \int_0^{\infty} (x^T Q x + \rho u^T R u) dt \quad (\text{eqn 2.3})$$

where R is positive definite ($R > 0$) for bounded input and ρ is a scalar. Q is a semi-positive definite matrix ($Q \geq 0$).

When both Q and R are diagonal matrices, ρ defines the relative weight between the state and control weighting matrices in the performance index. The quadratic form $x^T Q x$ and $u^T R u$ provide a weighted measure of the magnitude of the states and control vector respectively. The steady state control law that minimizes J is given by,

$$u(t) = -F x(t) \quad (\text{eqn 2.4})$$

where F is the feedback gain matrix which is given by

$$F = -R^{-1} B^T P \quad (\text{eqn 2.5})$$

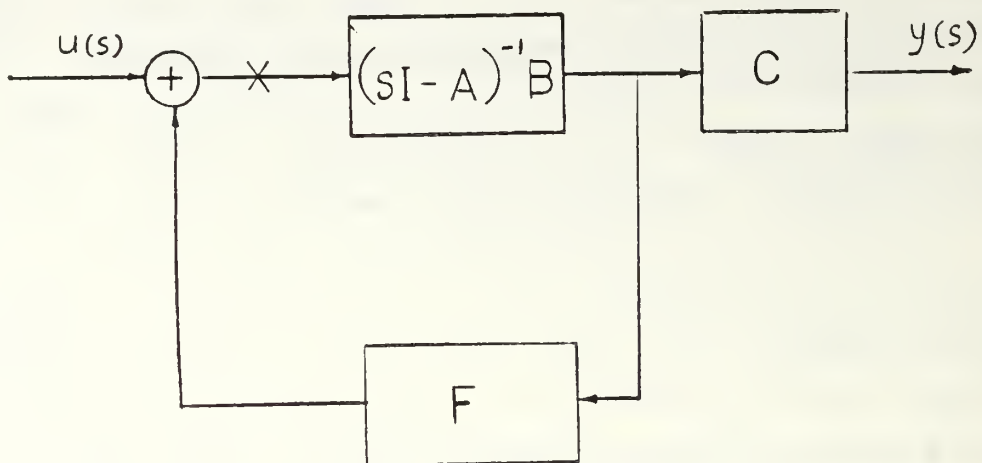
The positive definite matrix P is given by the solution to the steady state Riccati equation.

$$P A + A^T P + Q - P B R^{-1} B^T P = 0 \quad (\text{eqn 2.6})$$

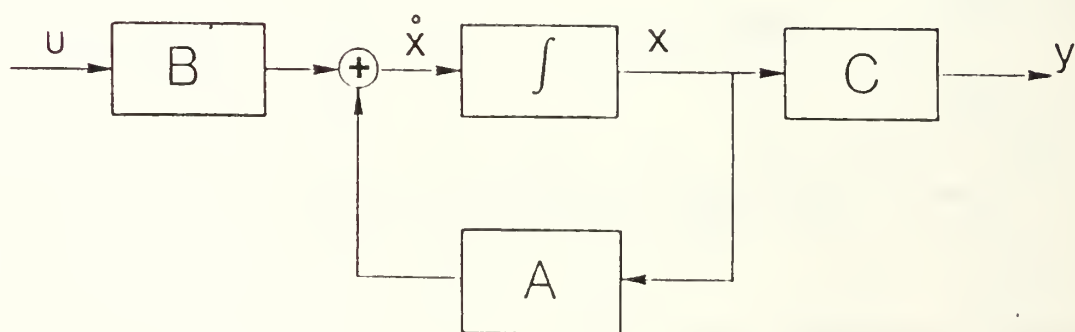
Equation 2.5 and 2.6 are well-known results in optimal control theory that yield the optimal closed-loop system

$$\dot{x}(t) = [A + B F] x(t) \quad (\text{eqn 2.7})$$

whose closed-loop poles are given by the eigenvalues of the matrix $[A + B F]$. The LQ system given by equations 2.1 through 2.6 is closed-loop stable and can be represented by the feedback configuration in both the time and frequency domain as shown in Figure 2.1 .



Frequency Domain



Time Domain

Figure 2.1 State Feedback System-Time and Frequency Domain.

In Figure 2.1, if the loop is broken at the input (X) as shown, the loop transfer function is given by

$$G(s) = F(sI - A)^{-1}B \quad (\text{eqn 2.8})$$

The matrix $[I + G(s)]$ is called the Return Difference Matrix and will be shown to have some important feedback properties in the following section.

B. POLE ASSIGNMENT PROBLEM

In the most general form, the state feedback pole assignment problem in control system design can be formulated precisely as follows:

"Given real matrices (A,B) of order (nxn, nxm) respectively and a set of n complex numbers $(\lambda_1, \lambda_2, \dots, \lambda_n)$ closed under complex conjugation. Find a real mxn matrix F such that the eigenvalues of $[A+BF]$ are λ_i , $i = 1, 2, 3 \dots n$."

In general, the closed-loop eigenvalues of (A+BF) can be arbitrary located in the complex plane, with the only restriction that complex characteristic eigenvalues must occur in complex conjugate pair. In other words, if the matrices (A,B) are a completely controllable pair, the stability of the system can always be improved by state feedback. If the (A,B) pair is not completely controllable, then the system is required to be 'stabilizeable' meaning that in its controllability canonical form given by equation 2.9 below, the matrix A_{22} is asymptotically stable or any unstable subspace of equation 2.9 is also in the controllable subspace.

$$\begin{bmatrix} \dot{x}_1(t) \\ \dot{x}_2(t) \end{bmatrix} = \begin{bmatrix} A_{11} & A_{12} \\ 0 & A_{22} \end{bmatrix} \begin{bmatrix} x_1(t) \\ x_2(t) \end{bmatrix} + \begin{bmatrix} B \\ 0 \end{bmatrix} \begin{bmatrix} u_1(t) \\ u_2(t) \end{bmatrix}$$

(eqn 2.9)

The solution to the pole assignment problem in the single input ($m=1$) case, when it exists, can be shown to be unique. In the multiple input case ($1 < m < n$), the solution of the so-called inverse eigenvalue problem is, in general, undetermined with many degrees of freedom. Additional conditions must be supplied in order to eliminate the extra degrees of freedom. This has been an area of active research in recent years and a number of approaches have been developed to relate the extra degrees of freedom with properties such as system eigenvectors, transmission zeros and robustness [Refs. 16,17]. In this work, it is shown that Linear Quadratic formulation incorporating equation 2.3 will partly remove the uncertainty that exists in the multiple inputs case. It will be shown that useful properties like robustness are guaranteed. It is also shown that the LQ type of pole placement formulation, when combined with eigenvectors assignment type of formulation, will produce some very useful design and synthesis procedures. In the present work, however, only the LQ eigenvalue placement problem is addressed; the problem of combined eigenvalue and eigenvector assignment is briefly described in Chapter 5.

C. WEIGHTING MATRICES AND SYSTEM PERFORMANCES

This section briefly reviews the effect of weighting matrices on system performance. The physical aspects of the weighting matrix for both single and multiple input systems are presented first together with some discussion on the asymptotic behavior of the LQ system.

It was shown in the last section that under complete controllable conditions, the time-invariant linear system can be stabilized by a linear feedback control law. For the regulator-type problem where the aim is to bring the system from an arbitrary initial state to the zero state, the closed loop poles can be chosen far to the left on the complex plane. Convergence to the zero state is fast but the large input required may not be practical. This naturally leads to an optimization problem where the trade off is between speed of convergence to zero and the magnitude of the input. This is reflected in the two quadratic terms in the performance index. The quantity $x^T Q x$ in the first term of the performance index is a measure of the extent in which the state at time t deviates from the zero state. The matrix Q determines how much weight is attached to each of the component of the state. The integral $\int (x^T Q x) dt$ is a criteria for the cumulative deviation of $x(t)$ from the zero state during the interval.

The problem of large control input is resolved by incorporating the second quadratic term, $\int (u^T R u) dt$. Larger value in the element of the control weighting matrix R will result in smaller input. It will be shown later that one can manipulate R to achieve some secondary design objectives. The remaining parameter ρ , that need to be specified, accounts for the relative weighting between the state and the control inputs. Selecting optimal value for ρ depends on the particular problem and the design requirement. As ρ

decreases, the integrated square regulating error decreases but the integrated square input increases. Very often, the optimal control problem is solved for many different values of ρ . A graph similar to that shown in figure 2.2 is obtained where the integrated square regulating error is plotted versus the integrated square input. An appropriate value of ρ is then selected to give sufficiently small regulator error without excessively large input.

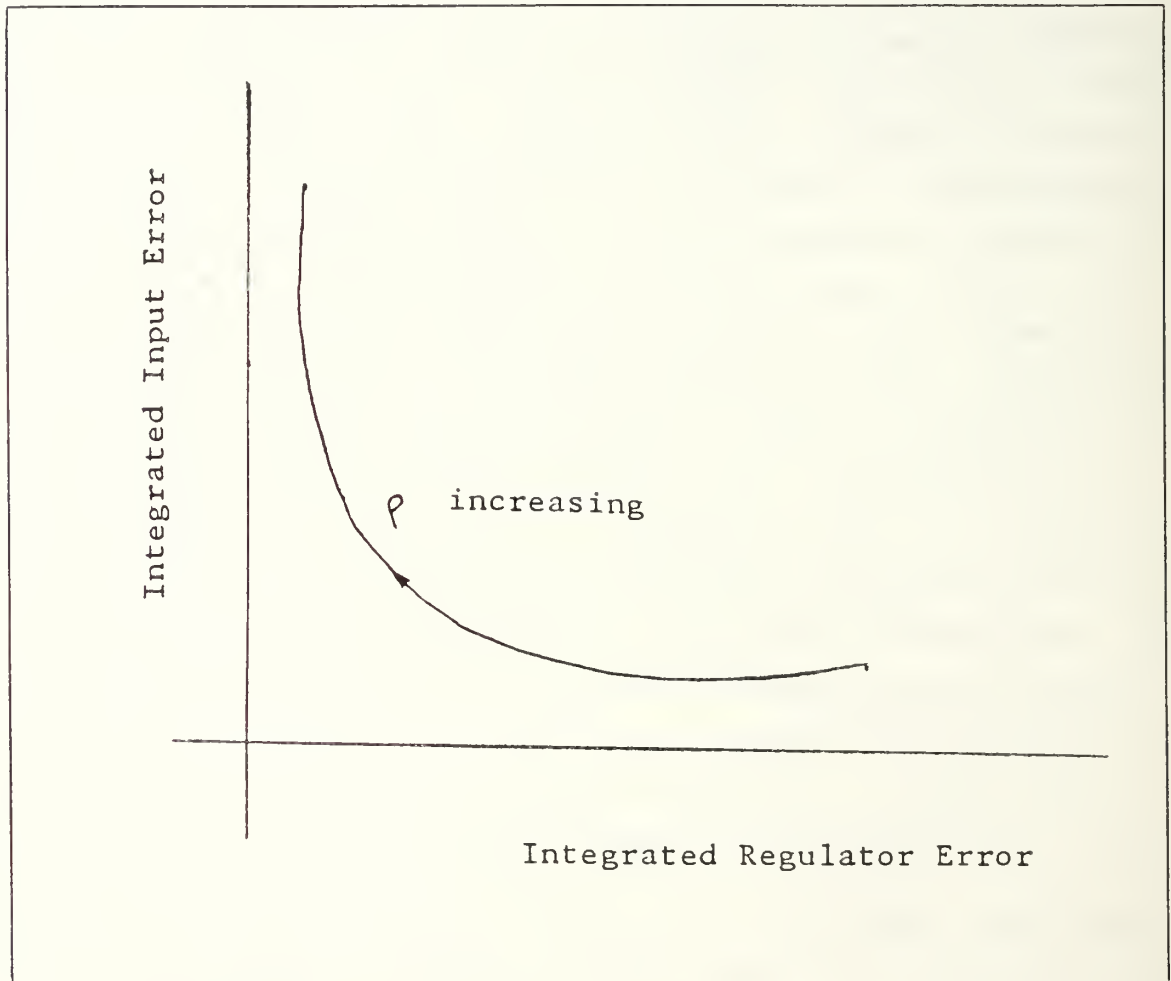


Figure 2.2 Selecting Relative Weighting Parameter ρ .

The special case where ρ approaches zero ,the so-called asymptotic properties, has been shown to provide good insight for LQ control system design. Some of the results from [Refs. 4,18] are summarized below.

As ρ decreases to zero for the system given by equations 2.1 through 2.6, some (say q) of the closed-loop poles go to infinity while other ($n-q$) stay finite. Those remaining finite approach the left half plane zero of $\det[B^T(-sI-A^T)^{-1}Q(sI-A)^{-1}B]$. If m is the dimension of the input vector, then at least m closed-loop poles approaches infinity. All closed-loop poles that go to infinity do so by grouping into several Butterworth patterns of different order and radii. For ρ approaching infinity, the closed-loop poles approach the mirror image of the plant open-loop poles. To illustrate the asymptotic concept, two examples from [Ref. 4] are given below.

An example of the asymptotic properties of a single input system is shown in Figure 2.3 . The closed-loop poles of a position control system using LQ type feedback are plotted as a function of ρ . The system has two open-loop poles at -4.6 and 0.0 . As ρ decreases, the closed-loop poles go to infinity along two straight lines that make an angle of $\pi/4$ with the negative real axis. As ρ approaches infinity, both closed-loop poles first meet on the negative real axis and then approach the open-loop poles at (-4.6, 0).

Figure 2.4 shows the asymptotic loci of the closed-loop poles for a multiple input system where $n=4$ and $m=2$. There are four open-loop poles at $(-0.006123, \pm j0.09353)$ and $(-1.250, \pm j1.394)$ and both the state and control weighting matrices are taken to be of a diagonal form. As ρ approaches to zero, one closed-loop pole stay finite at -1.002. The remaining three go to infinity, two of which assume a second order Butterworth pattern and the last one

approaches on the negative real axis. When ρ approaches infinity, all closed-loop poles approach the open-loop poles.

Many researchers have explored this asymptotic properties. Stein [Refs. 9,10] developed a procedure to select the control weighting matrices for a desired closed-loop asymptotic eigenstructure. It will be shown in the subsequent Chapters that the asymptotic properties provide useful guideline for the design procedures to be described.

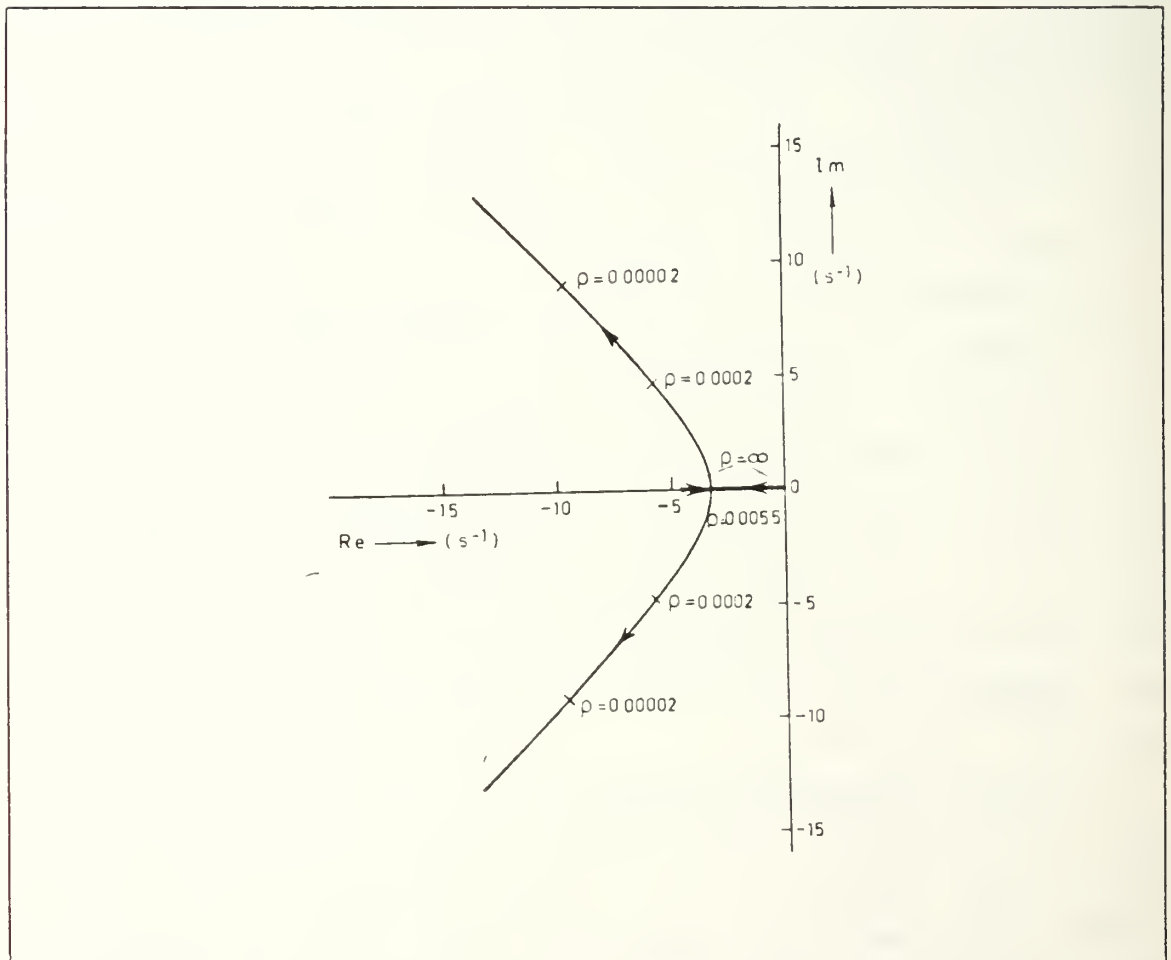
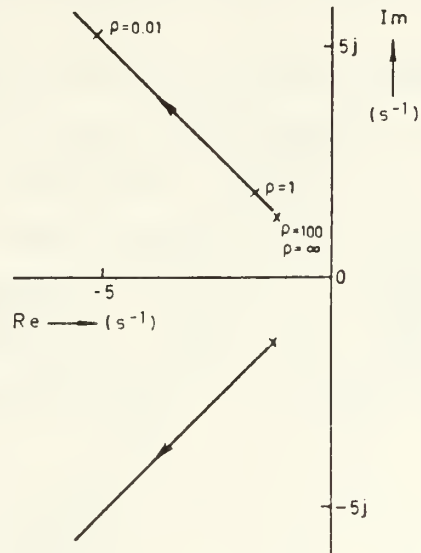
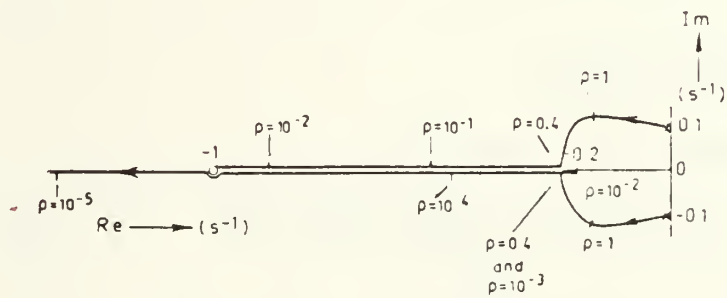


Figure 2.3 Single Input Asymptotic Root Loci.



Faraway Poles



Nearby Poles

Figure 2.4 Multiple Input Asymptotic Root Loci.

III. ROBUSTNESS THEORY AND LINEAR QUADRATIC CONTROL

In the last Chapter, the effect of weighting matrices on the closed-loop pole of the LQ system was discussed. This Chapter addresses yet another important feedback property that control system designers are concerned with: Robustness. Robustness theory was developed when it was realized that the classical single loop Nyquist test was not adequate to guarantee stability when the multivariable open loop plant deviates from its model due to a variety of reasons [Ref. 15]. In the following section, the concept of robust design for both SISO and MIMO general feedback system is reviewed. The singular value analysis is discussed in term of multiplicative type of disturbances. Finally, robustness for linear quadratic state feedback are presented in terms of the effect of weighting matrices on the singular value curve.

A. ROBUSTNESS CONCEPTS

The concept of robustness for SISO system can best be described in terms of the definition of phase and gain margin. A system characterized by good gain and phase margin implies that changes in the plant model parameters and changes in the loop gain and/or phase may be accommodated without loss of stability. The gain and phase margins of a SISO system are defined with reference to the perturbed system in Figure 3.1

Assuming that the unperturbed system ($l(j\omega)=1$) is stable, the positive (or negative) phase margin is the value of ϕ greater (or less) than zero at which the perturbed system with $l(j\omega) = k \cdot \exp(j\phi)$ becomes unstable. The upward

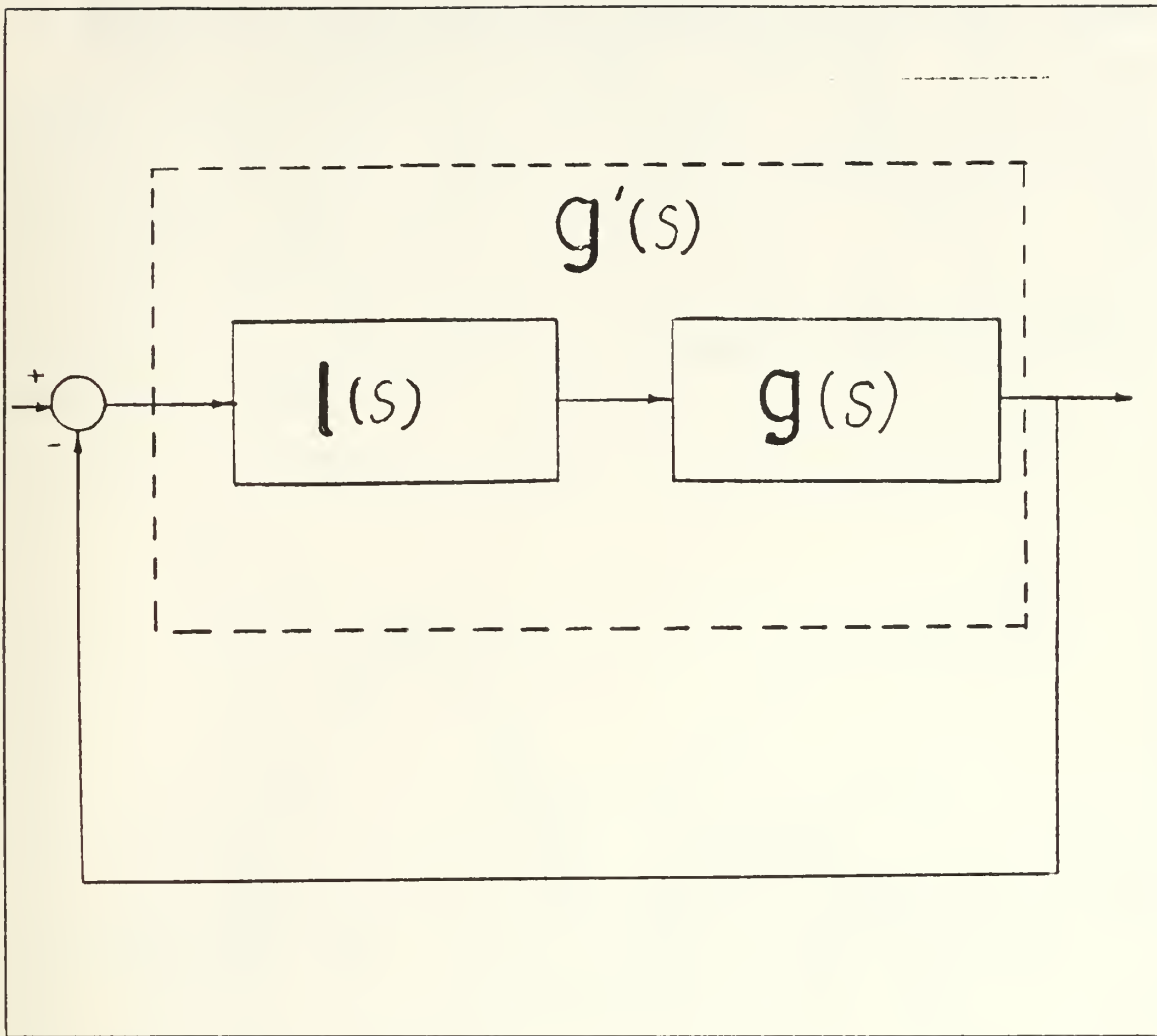


Figure 3.1 The Perturbed SISO System.

(or downward) gain margin is the smallest (or greatest) value of $l(j\omega)=k=\text{constant}$, for which the system become unstable. With reference to the classical Nyquist plot in Figure 3.2 , gain and phase margin are defined as,

$$GM \uparrow (\text{upward}) = 1/k_1 \qquad GM \downarrow (\text{downward}) = 1/k_2$$

$$PM^+ = \alpha_1 \qquad PM^- = \alpha_2$$

A set of minimum guaranteed GM and PM may be obtained by defining $\alpha_o = \min[1 + g(j\omega)]$ as shown in Figure 3.3 where,

$$GM = 1/(1 \pm \alpha_o)$$

and

$$PM = \pm \cos^{-1}(1 - \alpha_o^2/2)$$

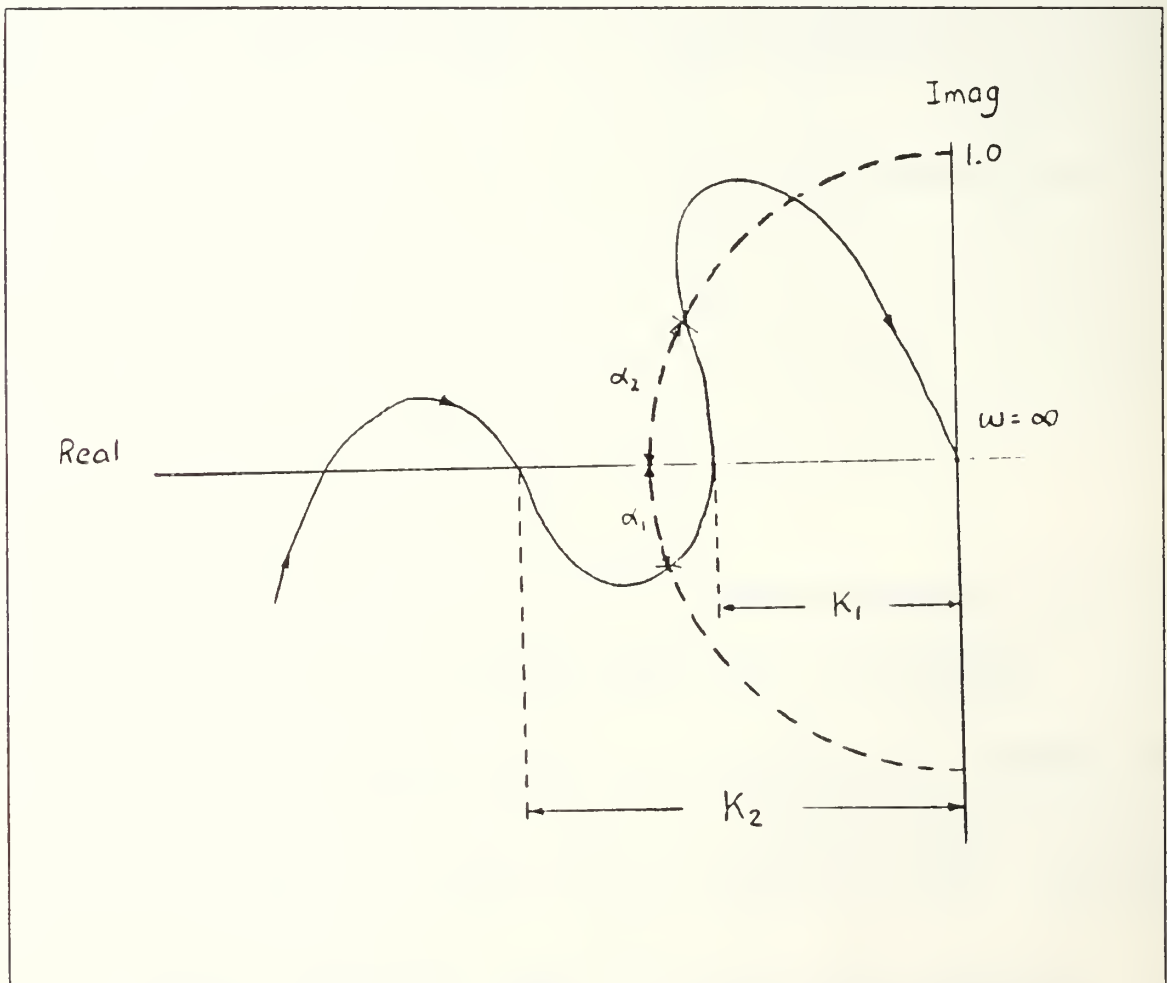


Figure 3.2 The Nyquist Plot - Gain and Phase Margin.

Note that in the above definition of GM and PM, either ϕ_i 'or' κ_i is allowed to change in the loop. The allowed changes are therefore very restrictive. A more useful definition of the gain and phase margin for MIMO system that accounts for simultaneous changes in both ϕ_i and/or κ_i (the so-called universal gain and phase margin) has been derived in [Ref. 19]. From Figure 3.3 it can be seen that α_0 may provide a rather conservative estimate of the actual gain and phase margin.

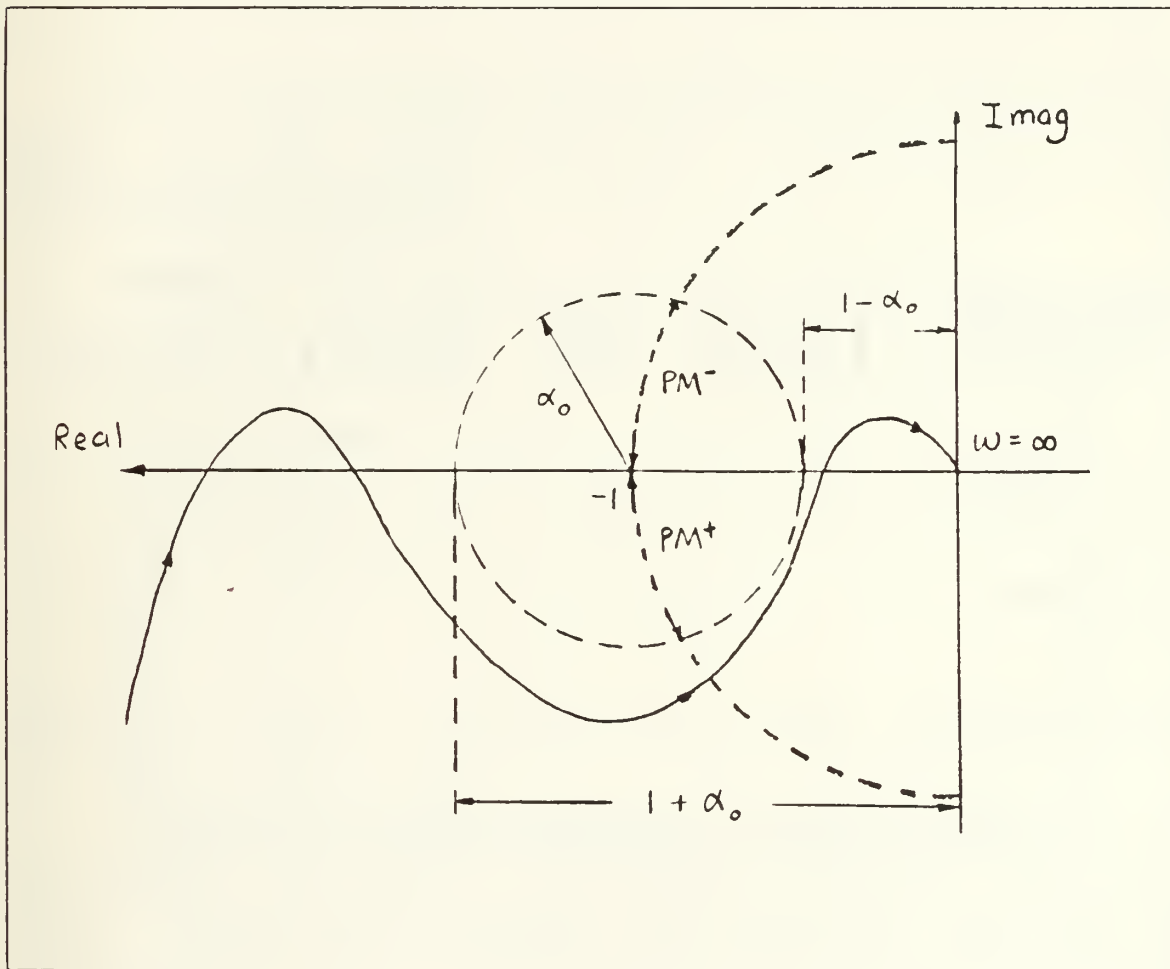


Figure 3.3 The Minimum Guaranteed GM and PM.

In MIMO system, gain and phase margin characterize the ability of the system to tolerate gain and/or phase changes within all loops simultaneously. Figure 3.4 shows a perturbed MIMO system with the assumption that $L(j\omega) = \text{Dia}(l_1(j\omega), l_2(j\omega) \dots)$. As in the SISO case, the system will remain stable as long as $l_i(j\omega)$ satisfy $GM^- < k_i < GM^+$ (assuming that ϕ_i 's are zero). For the case when the magnitude of $l(j\omega)$ are constant, the system will remain stable for all ϕ_i 's that satisfy $PM^- < \phi_i < PM^+$.

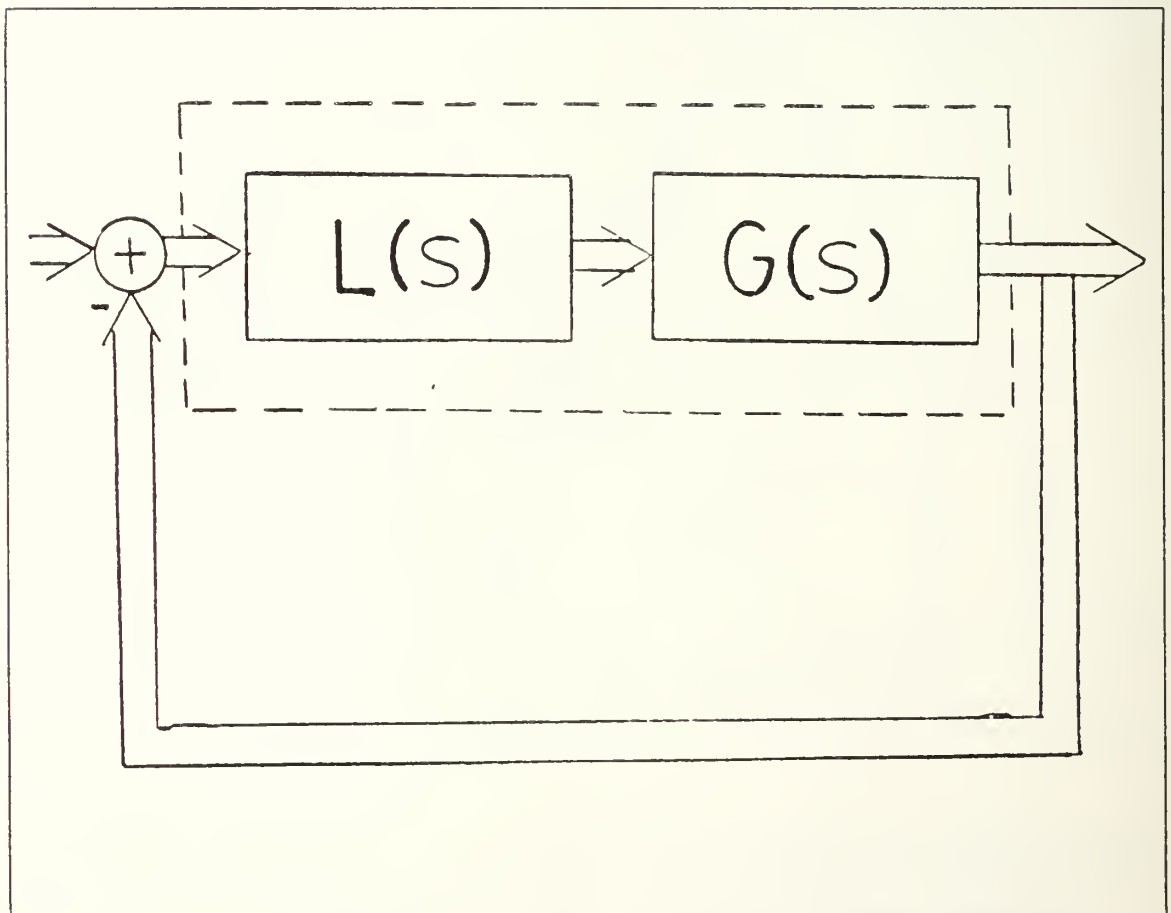


Figure 3.4 The Perturbed MIMO System.

The set of minimum guaranteed multivariable gain and phase margins are determined by the minimum singular value ($\underline{\sigma}$) of the return difference matrix $[I + G(s)]$, where $G(s)$ is the open loop transfer matrix and consists of the plant and its controller. Note the similarity between the SISO and MIMO cases,

α_o : nearness of $(1 + g(s))$ to the origin.

$\underline{\sigma}$: nearness of matrix $[I + G(s)]$ to singularity.

Two important results that are related to multivariable phase and gain margin developed in [Ref. 15] are next presented as a theorem.

THEOREM: The multiplicative perturbed system (Figure 3.4) is stable if either of the following conditions hold:

$$1. \quad \underline{\sigma} [I + G(s)] > \bar{\sigma} [L^{-1}(s) - I] \quad (\text{eqn 3.1})$$

$$2. \quad \underline{\sigma} [I + G^{-1}(s)] > \bar{\sigma} [L(s) - I] \quad (\text{eqn 3.2})$$

where $\bar{\sigma}$, $\underline{\sigma}$ denote the maximum and minimum singular values of $[I+G(s)]$ respectively.

It will be shown that condition 1 can be related to the optimality condition of LQ system and hence provides a useful relationship between the weighting matrices and robustness.

B. ROBUSTNESS IN LQ SYSTEM

Robustness properties pertaining to LQ system are closely related to the frequency domain optimality condition. For the SISO case, Kalman [Ref. 1] showed that the return difference transfer function satisfies the inequality

$$(1 + g(j\omega)) \geq 1 \quad (\alpha_0 = 1) \quad (\text{eqn 3.3})$$

Inspection of the Nyquist diagram in Figure 3.5 clearly indicates that a SISO LQ state feedback has a guaranteed infinite upward gain margin, 0.5 downward gain margin and a minimum phase margin of ± 60 deg.

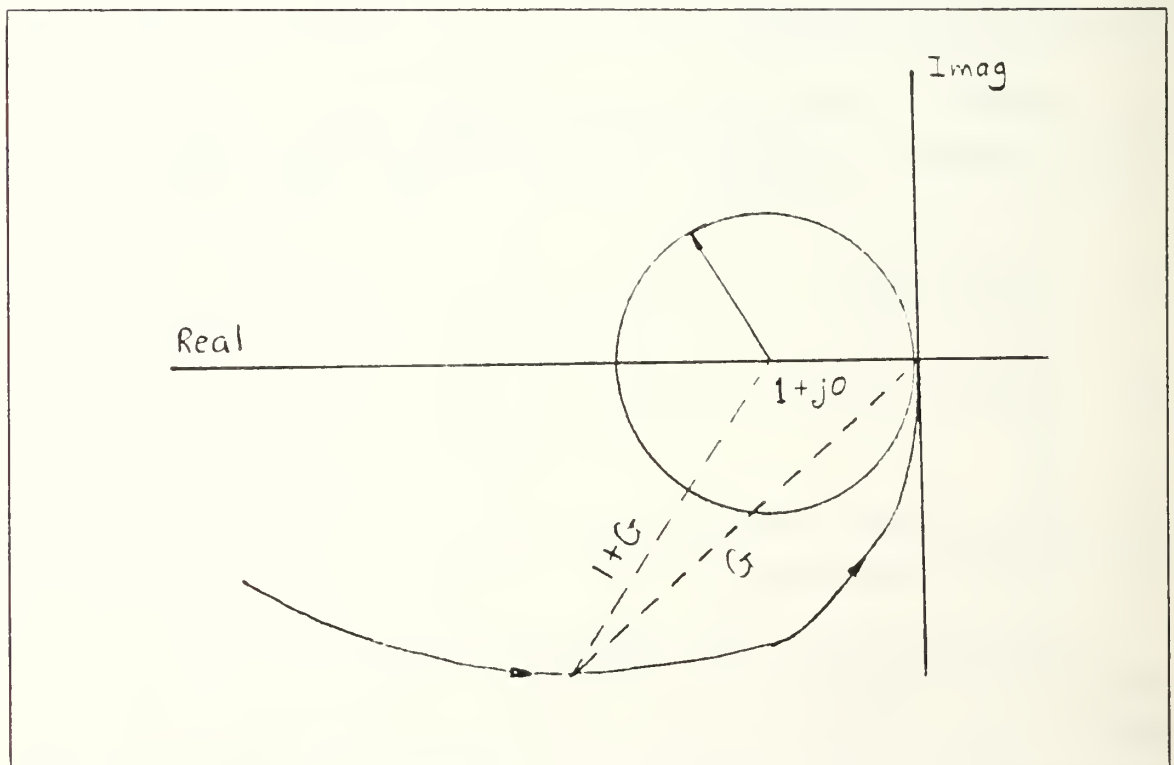


Figure 3.5 SISO LQ Nyquist.

In a similar manner, multivariable stability margin can be derived from the multivariable version of condition 1 in equation 3.1 ;

$$[I + G(s)]^H R [I + G(s)] > R \quad (\text{eqn 3.4})$$

where

$$G(s) = R^{-1} B^T P [sI - A]^{-1} B$$

and $P \geq 0$ satisfies the steady state Riccati equation.

It can be shown that condition 1 in equation 3.1 above can be written as

$$\underline{\sigma} [I + R^{1/2} G(j\omega) R^{-1/2}] \geq 1 \quad (\text{eqn 3.5})$$

which can be reduced to

$$\underline{\sigma} [I + G(j\omega)] > 1 \quad (\text{eqn 3.6})$$

where R is diagonal (i.e. $R = \rho I$ for some positive scalar ρ). Like the SISO case, the multivariable LQ regulator with loop transfer matrix $G(s)$ that satisfies equation 3.5 and 3.6 has, in all the feedback loops, a guaranteed minimum gain and phase margin given by,

$$GM = 1/2, \text{ and } PM = \pm 60 \text{ deg}$$

The case when R is not diagonal is interesting as condition 3.6 no longer apply and additional trade-off can be obtained by including off diagonal terms in the control input weighting matrix. This is especially useful when the nature and structure of the disturbances are known. Design involving the off diagonal R matrix will be shown in Appendix A.

IV. DESIGN PROCEDURE

A. GENERAL

The background material described in Chapter 2 and 3 are the basis for developing a computer aided design package and the corresponding design procedure. The design philosophy presented in this Chapter is most useful when a designer is able to characterize the desired system in terms of closed-loop eigenvalues and time response. Some initial physical insight of the weighting matrices, their asymptotic properties and the nature of the perturbation will be useful in the design process. This Chapter begins with a discussion of the various approaches to the LQ pole placement problem. The selected approach and the corresponding computer aided design package is presented. The design philosophy and design procedure for both the reduced order and full order model are then given.

B. APPROACHES

All the pole placement algorithms for LQ system that have been developed so far require an expression that relate the characteristic equation of the optimal system with the elements of the weighting matrices. Two such formulations for the stabilizeable and detectable time-invariant linear system (equation 2.1) and the quadratic criteria (equation 2.3) are given by

$$\phi_c(s)\phi_c(-s) = \phi(s)\phi(-s)\det[I + R^{-1}H^T(-s)QH(s)] \quad (\text{eqn 4.1})$$

and

$$\phi_c(s)\phi_c(-s) = \phi(s)\phi(-s)\det[I + 1/\rho R^{-\frac{1}{2}} H^T(-s)QH(s)R^{-\frac{1}{2}}] \quad (\text{eqn 4.2})$$

where $\phi_c(s) = \det[sI - A + BF]$ and $\phi(s) = \det[sI - A]$ are the closed-loop and open-loop characteristic polynomials respectively. $H(s) = C(sI - A)^{-1}B$ is the open loop transfer matrix of the system.

Both formulations have been used in root-locus type of procedure to investigate how the closed-loop poles move as weighting matrices changes [Refs. 5,6]. For MIMO case, there has been little success due to problems involving polynomial matrices. In this work, a different approach is adopted. Equation 4.1 or 4.2 is formulated as a numerical optimization problem in which the objective function is made equal to the determinant part of equation 4.1 or 4.2

$$\text{Obj} = \det[I + R^{-1}H^T(s)QH(s)] \quad (\text{eqn 4.3})$$

or

$$\text{Obj} = \det[I + 1/\rho R^{-\frac{1}{2}} H^T(-s)QH(s)R^{-\frac{1}{2}}] \quad (\text{eqn 4.4})$$

For a given desired closed loop pole $s = s_d$, equation 4.3 or 4.4 becomes

$$\phi_c(s_d)\phi_c(-s_d) = 0 \quad (\text{eqn 4.5})$$

Providing that $\phi(s_d)\phi(-s_d)$ is not equal to zero, the objective function must equal to zero if the particular Q, R set is to correspond to the desired closed-loop poles. Convergence to zero for a given set of Q and R is therefor automatically guaranteed.

The pole placement problem can therefore be solved as an unconstrained multivariable optimization in which the elements of Q and R are varied to make the objective

function in equation 4.3 and 4.4 approach zero. This was done during the early phase of the work. It was later discovered that more insight to the problem can be obtained by first transforming the problem to an appropriate coordinate system and then to perform pole placement one at a time. This is of advantage as a system designer is often satisfied with several open-loop poles in a large system. Reassigning poles in the the reduced-order model will reduce computer time and memory requirement.

The optimization routine selected for this work is the so-called SUMT method (Sequential Unconstrained Minimization Techniques) obtained from the ADS package in [Refs. 20,21]. In this method, the objective function (eg. equation 4.3 or 4.4) and any constraint equation are formulated into an augmented objective function in which the problem is solved as an unconstraint optimization task.

C. POLE PLACEMENT ALGORITHMS

For ease of implementation and better insight, only the problem of determining the state weighting matrix Q (given R) that gives a set of closed-loop eigenvalues is considered. It must be emphasised that the algorithm can also be formulated to determine R (for a given Q) or to vary Q and R at the same time. In most cases, the present formulation is adequate as designers usually have some knowledge about the control weighting matrix. System matrix that has real and distinct eigenvalues is presented first, follows by cases where A has complex eigenvalues and repeated eigenvalues.

1. System with Real and Distinct Eigenvalues

The original system given by equation 2.1 and 2.2 is first transformed into a diagonal form using the transformation given by;

$$\mathbf{x}(t) = \mathbf{M}\mathbf{z}(t)$$

(eqn 4.6)

where \mathbf{M} is an eigenvector matrix corresponding to the system matrix \mathbf{A} . and $\mathbf{z}(t)$ is the new state vector. The transformed system in the new coordinate is given by,

$$\dot{\mathbf{z}}(t) = \mathbf{\Lambda} \mathbf{z}(t) + \mathbf{M}^{-1}\mathbf{B}\mathbf{u}(t)$$

(eqn 4.7)

where $\mathbf{\Lambda}$ is a diagonal matrix $\text{dia}[\lambda_1, \lambda_2, \lambda_3, \dots]$

The performance index (equation 2.3), when expressed in terms of the new state vector $\mathbf{z}(t)$ becomes,

$$\begin{aligned} J &= \int_0^{\infty} (\mathbf{x}^T \mathbf{Q} \mathbf{x} + \mathbf{u}^T \mathbf{R} \mathbf{u}) dt \\ &= \int_0^{\infty} (\mathbf{z}^T \mathbf{M}^T \mathbf{Q} \mathbf{M} \mathbf{z} + \mathbf{u}^T \mathbf{R} \mathbf{u}) dt \\ &= \int_0^{\infty} (\mathbf{z}^T \hat{\mathbf{Q}} \mathbf{z} + \mathbf{u}^T \mathbf{R} \mathbf{u}) dt \end{aligned}$$

(eqn 4.8)

where $\hat{\mathbf{Q}} = \mathbf{M}^T \mathbf{Q} \mathbf{M}$.

It can be shown that to move an open-loop pole to its new location given by $s_i = s_d$, only \hat{Q}_i is required and other \hat{Q}_s have no effect on the pole assignment. As an example, to move the open-loop pole at $s = \lambda_2$ for $\mathbf{\Lambda} = \text{dia}[\lambda_1, \lambda_2, \lambda_3, \dots]$ to its new location $s = \bar{\lambda}_2$, only $\hat{\mathbf{Q}} = \text{dia}[0, \hat{q}_2, 0, \dots]$ is required.

$\hat{\mathbf{Q}}$ can then be selected according to equation 4.3 or 4.4 using the optimization routine. As currently implemented in the program, there is no constraint equation formulation. Once the value of $\hat{\mathbf{Q}}$ that satisfies the desired pole location is obtained, the system is transformed back to the original coordinate system via,

$$\mathbf{Q} = \mathbf{M}^{-T} \hat{\mathbf{Q}} \mathbf{M}^{-1}$$

(eqn 4.9)

With Q known, and R given, the optimal feedback gain F can be obtained by solving the steady state Riccati equation as given in equation 2.5 and 2.6 . Since the pole placement is done in the decoupled coordinate system, only the eigenvalues that correspond to Q_i is reassigned; all other eigenvalues remain unchange. It can be also shown that the eigenvector corresponding to an eigenvalue is also unchange, this property will be shown to be useful in the reduced order formulation of the linear quadratic problem.

If desired, the problem here can also be formulated to move more than one eigenvalues in one run. This can be done by modifying the objective function to include more terms as follows;

$$Obj = \sum_{i=1}^n \det[I + R^{-1} H^T(-s)QH(s)R^{-1}] \quad (eqn 4.10)$$

where n is the number of poles to be reassigned.

The augmented matrix $A_{aug} = [A + BF]$ is then computed. If desired, the procedure may be repeated to move other open loop eigenvalue to its specified position using the new A_{aug} as the starting plant matrix. This will in turn result in another set of Q and F . The effective Q_e and F_e after n reassignments are given by

$$Q_e = Q_1 + Q_2 + \dots Q_n \quad (eqn 4.11)$$

and

$$F_e = F_1 + F_2 + \dots F_n \quad (eqn 4.12)$$

The above pole assignment procedure can also be applied to an optimal system where an initial starting Q and R are given. A good example is when the control system designer has some knowledge of the weighting matrix but would also like to meet a specific time response requirement.

2. System with Complex Eigenvalues

If the same similarity transformation mentioned in the last section is used, the transformation matrix will be complex. To be able to work with real matrix, an auxiliary transformation of the form given by equation 4.13 is used;

$$x(t) = Tz(t) \quad (\text{eqn 4.13})$$

$T = ML$ and M is the eigenvector matrix (equation 4.6) The matrix L is given by,

$$L = \begin{bmatrix} 0.5 & -0.5j & 0.0 & \dots & 0.0 \\ 0.5 & 0.5j & 0.0 & \dots & . \\ 0.0 & 0.0 & 1.0 & \dots & . \\ . & \dots & \dots & 1.0 & . \\ 0.0 & \dots & \dots & \dots & 1.0 \end{bmatrix} \quad (\text{eqn 4.14})$$

The transformed system is then given by,

$$\dot{z}(t) = \Lambda Z(t) + T^{-1}Bu(t) \quad (\text{eqn 4.15})$$

with performance index ,

$$J = \int (z^T T^T Q Tz + u^T R u) dt \quad (\text{eqn 4.16})$$

where \hat{Q} is now given by $\hat{Q} = T^T Q T$.

It can be shown that to move a pair of complex eigenvalues given by $s = a + bj$, a weighting matrix of the form

$$\hat{Q} = \begin{bmatrix} 0 & 0 & 0 & \dots & 0 \\ 0 & \hat{q}_i & 0 & & . \\ 0 & 0 & \hat{q}_{i+1} & & . \\ 0 & . & . & & . \\ 0 & . & . & \dots & 0 \end{bmatrix}$$

(eqn 4.17)

with $\hat{q}_i = \hat{q}_{i+1}$ is required.

In a similar manner, \hat{Q} can be obtained by using the optimization routine, with the condition $\hat{q}_i = \hat{q}_{i+1}$ formulated as a constrained equation. Inverse transformation and determination of Q_e and F_e are identical to the distinct eigenvalue case with M in equation 4.9 replaced by T .

3. System With Repeated Eigenvalues

In this case, the system matrix cannot be diagonalized but the general procedure given above still apply. The system is first transformed into the Jordan canonical form ;

$$J = U^{-1} A U \quad (\text{eqn 4.18})$$

where U is a transformation matrix which is not the eigenvector matrix M . An example of the Jordan form is given below,

$$J = \begin{bmatrix} \lambda & 1 & 0 \\ 0 & \lambda & 1 \\ 0 & 0 & \lambda \end{bmatrix}$$

(eqn 4.19)

The only difference with the two procedures mentioned above is that the pole reassignment has to begin at the bottom of each Jordan block. For example, in the system given above (equation 4.19), the first re-assignment will result in a new system given by equation 4.20

$$J = \begin{bmatrix} \lambda & 1 & 0 \\ 0 & \lambda & 1 \\ 0 & 0 & s_d \end{bmatrix}$$

(eqn 4.20)

Determination of Q and F using the optimization and the inverse transformation routines are identical to the distinct eigenvalues case with M in equation 4.9 replaced by U.

D. COMPUTER PROGRAM DESCRIPTION

The computer aided design package developed in this thesis is illustrated in Figure 4.1. The program is developed using the top-down approach with special purpose subroutines called by the main control program. The driver program supports three independent modes with the data entry portion common to all modes:

1. Data Entry : The system matrices A, B, C and/or F, Q, R etc are entered through a data file. Design variables such as desired poles locations, elements of matrices to be varied etc, are also specified through the input data file.
2. Pole Placement Mode : In this mode, an arbitrary set of closed-loop eigenvalues is assigned by selecting the appropriate state weighting matrix. As shown in Figure 4.1, the transformation matrices for various

cases are computed first. The pole placement is achieved using the numerical optimization routine described in the last section; Q is obtained and then inverse-transformed to the original co-ordinate system. If desired, the results can be used as an input to the Linear Quadratic Control Program.

3. Linear Quadratic Control Mode : This part of the program is adopted from the OPTSYS program. Given a set of weighting matrices Q and R and the system matrices A , B and C , it computes the steady state feedback gain F , closed-loop eigenvalues, etc.
4. Singular Value Analysis Mode : This portion of the program allows the designer to analyze various designs obtained from the two modes mentioned above in term of singular value vs frequency plot. The main part of this program is adopted from [Ref. 22].

The three modes of operation mentioned above may be used in any order to implement specific design objectives. A typical design process will involve runs alternating between the three modes until a compromise between primary and secondary design objectives is achieved. Record of a typical design run together with a complete listing of the main program and their non-standard subroutine are given in Appendix B and C.

E. DESIGN PHILOSOPHY

The Linear Quadratic constant state feedback design philosophy for the linear time-invariant model is illustrated in Figure 4.2. It assumes that the location of the eigenvalues and system time response are the main design objectives. These objectives can be achieved using the pole placement procedure developed in this thesis. Single or multiple reassignment can be made in one run. A number of

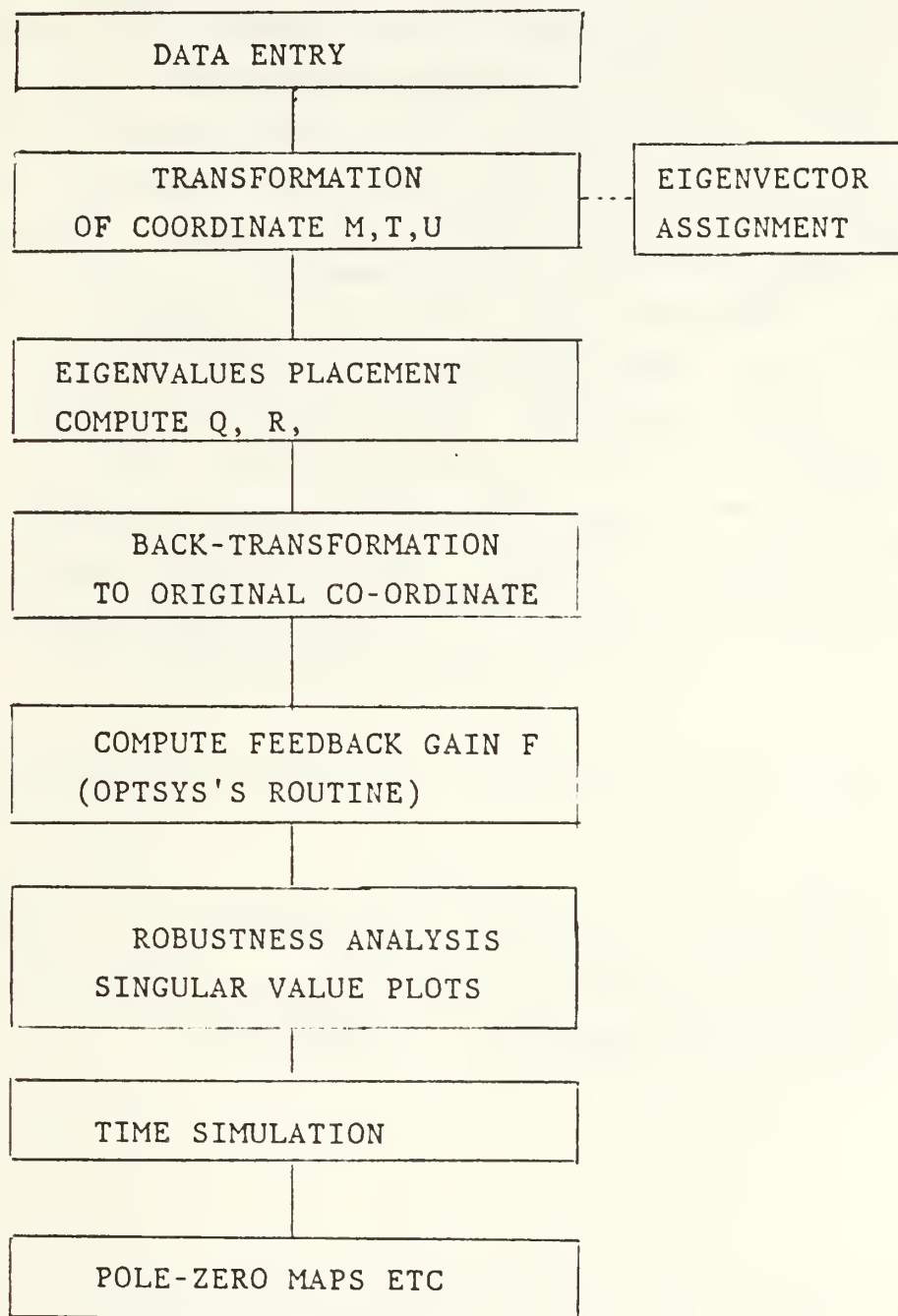


Figure 4.1 Computer Aided Design Package Organization.

designs can then be obtained using different starting control weighting matrices, different state weighting matrices and different assignment sequences. Physical constraints such as control input amplitude, control input energy as well as general system properties such as asymptotic behavior are heavily relied upon during this process. After the major objectives are satisfactorily achieved, secondary design objectives are considered. These include feedback gain reduction (by increasing the control weighting matrix), robustness (in terms of minimization of system sensitivity to modelling errors and/or parameter variations), zeros locations, eigenvectors assignments. The extra degrees of freedom available in the MIMO state feedback system can often provide a means to improve these secondary objectives while only slightly modifying the closed-loop pole assignment and thus the time response.

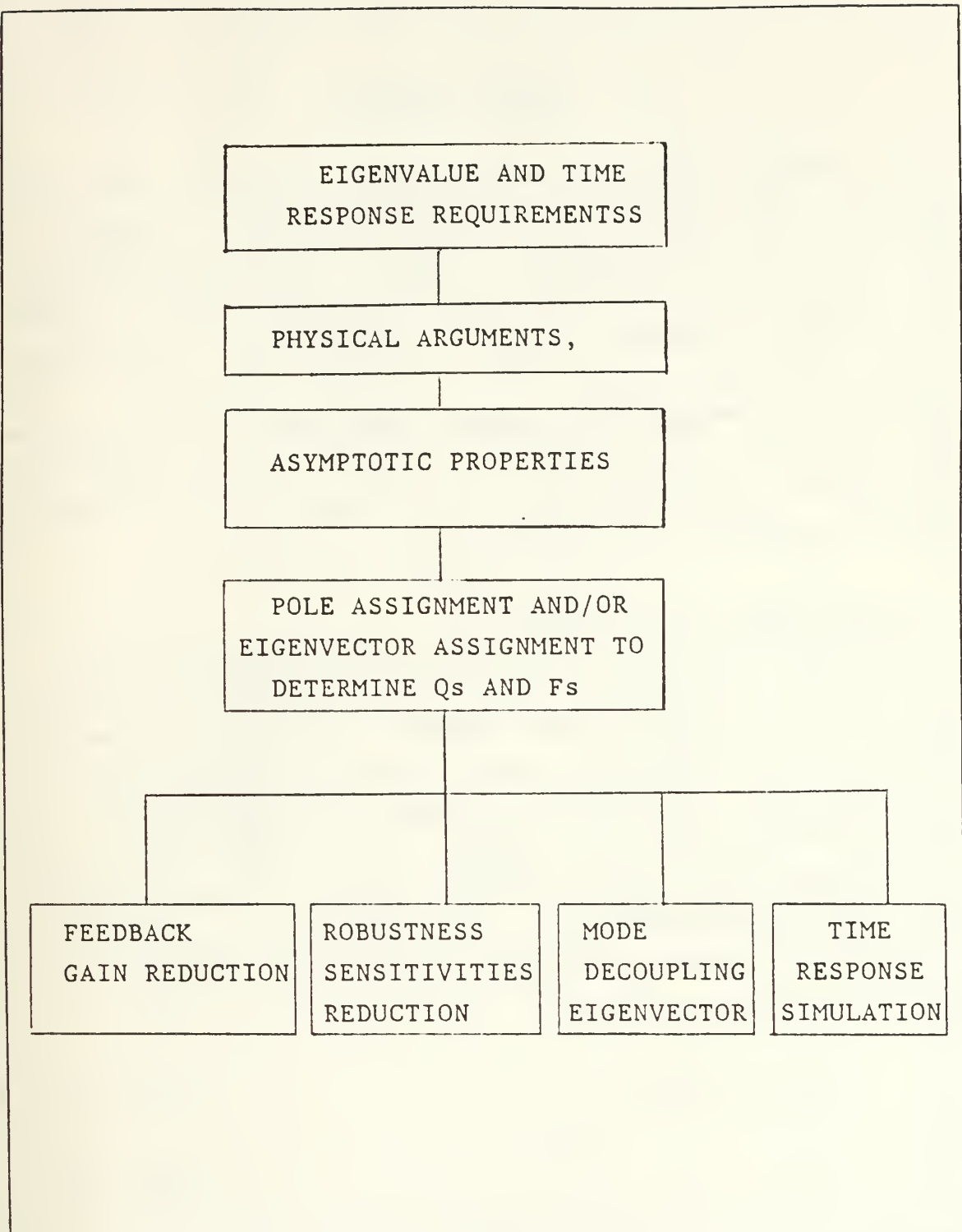


Figure 4.2 LQ Eigenvalue Assignment Philosophy.

V. DESIGN EXAMPLES

The design procedure described in Chapter 4 is illustrated in this Chapter by two design problems. A typical LQ structure and its properties, together with the pole placement procedure is first presented with a 2x2 model. A practical design problem is then presented for the highly coupled lateral channels of a CH-47 Helicopter. The resulting LQ designs are compared with other multivariable state feedback designs [Refs. 22,23]. It is shown that the procedure developed here is a viable tool for robust constant feedback controller design.

A. INTRODUCTORY 2X2 PROBLEM

This problem formulated in reference 23 serves to demonstrate how a highly cross-coupled multivariable control problem can be formulated and solved as a linear quadratic design problem, using the pole placement procedure. The problem provides excellent insight into the structure of the multivariable LQ system and its built-in robustness to cross-coupled perturbation.

Figure 5.1 shows a diagram of this basic 2X2 system in which the plant is given by

$$\dot{x}(t) = Ax(t) + Bu(t) \quad (\text{eqn 5.1})$$

$$y(t) = Cx(t) \quad (\text{eqn 5.2})$$

where

$$A = \begin{bmatrix} -1 & 0 \\ 0 & -1 \end{bmatrix}$$

$$B = \begin{bmatrix} 1 & 50 \\ 0 & 1 \end{bmatrix}$$

$$C = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

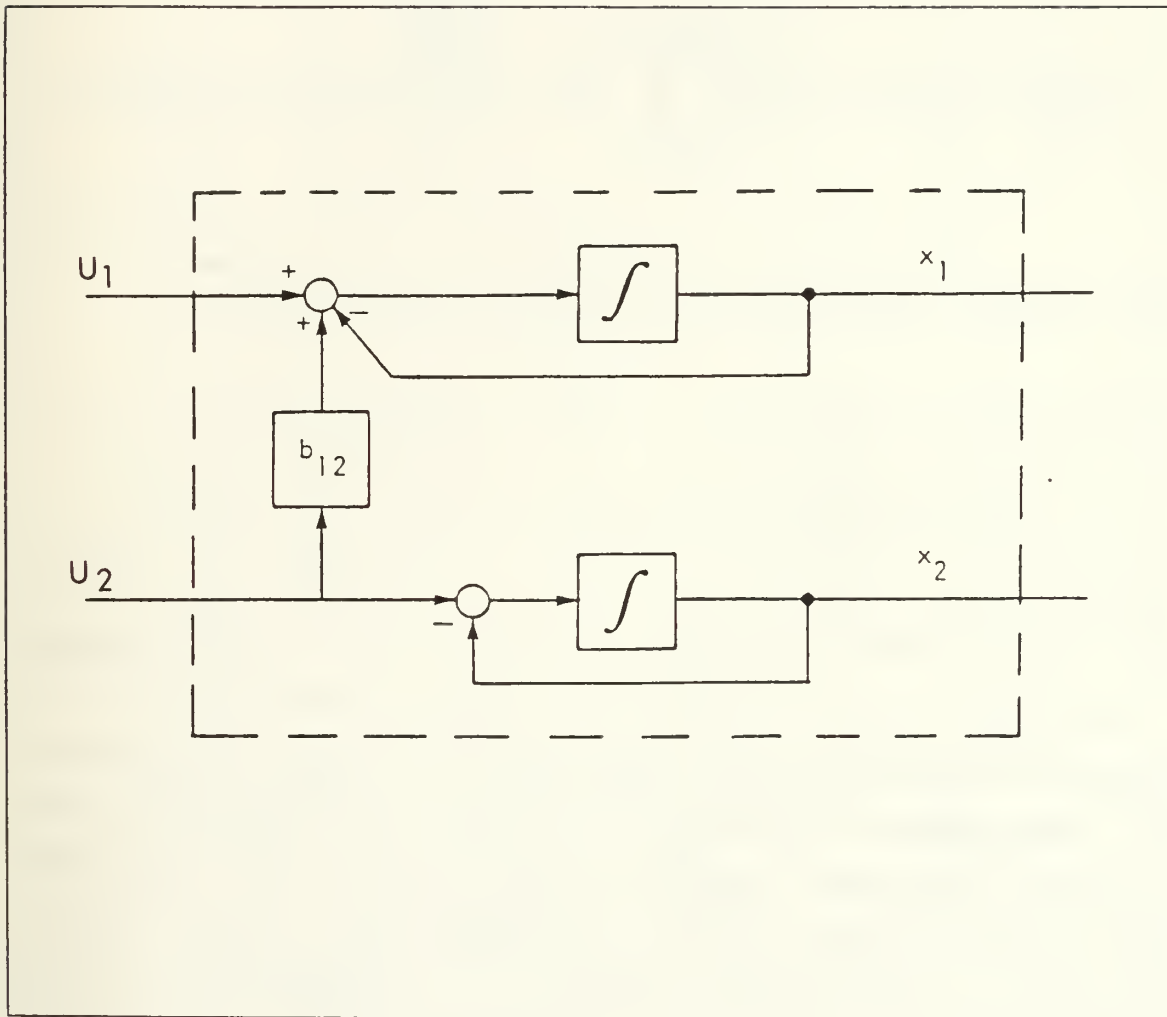


Figure 5.1 A Simple 2X2 MIMO Model.

This system has open-loop eigenvalue at -1, -1 and is therefore stable. The b_{12} term in the control matrix B is purposely made large to produce the cross-coupled effect from channel two to channel one(see Figure 5.1). The design requirement is to select a set of feedback gain F such that the closed-loop eigenvalues are at -2, -2. Assuming that this is the only requirement, it is not difficult to see that a unity feedback law of the form (Figure 5.2),

$$\begin{bmatrix} u_1 \\ u_2 \end{bmatrix} = \begin{bmatrix} -1 & 0 \\ 0 & -1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} u_{c1} \\ u_{c2} \end{bmatrix}$$

(eqn 5.3)

will produce the desired closed-loop system given by,

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \begin{bmatrix} -2 & 50 \\ 0 & -2 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} 1 & 50 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} u_{c1} \\ u_{c2} \end{bmatrix}$$

(eqn 5.4)

The above design seems to be acceptable as far as eigenvalues or time response is concerned. It is now shown that when robustness of the system is considered, the unity feedback gain controller performs rather poorly. On the other hand, design using LQ Pole Placement type of formulation will result in robust controllers. To demonstrate the lack of robustness of the unity feedback design, the feedback gain matrix F as given in equation 5.4 is perturbed slightly (by +5%) and the eigenvalues of the resulting closed-loop

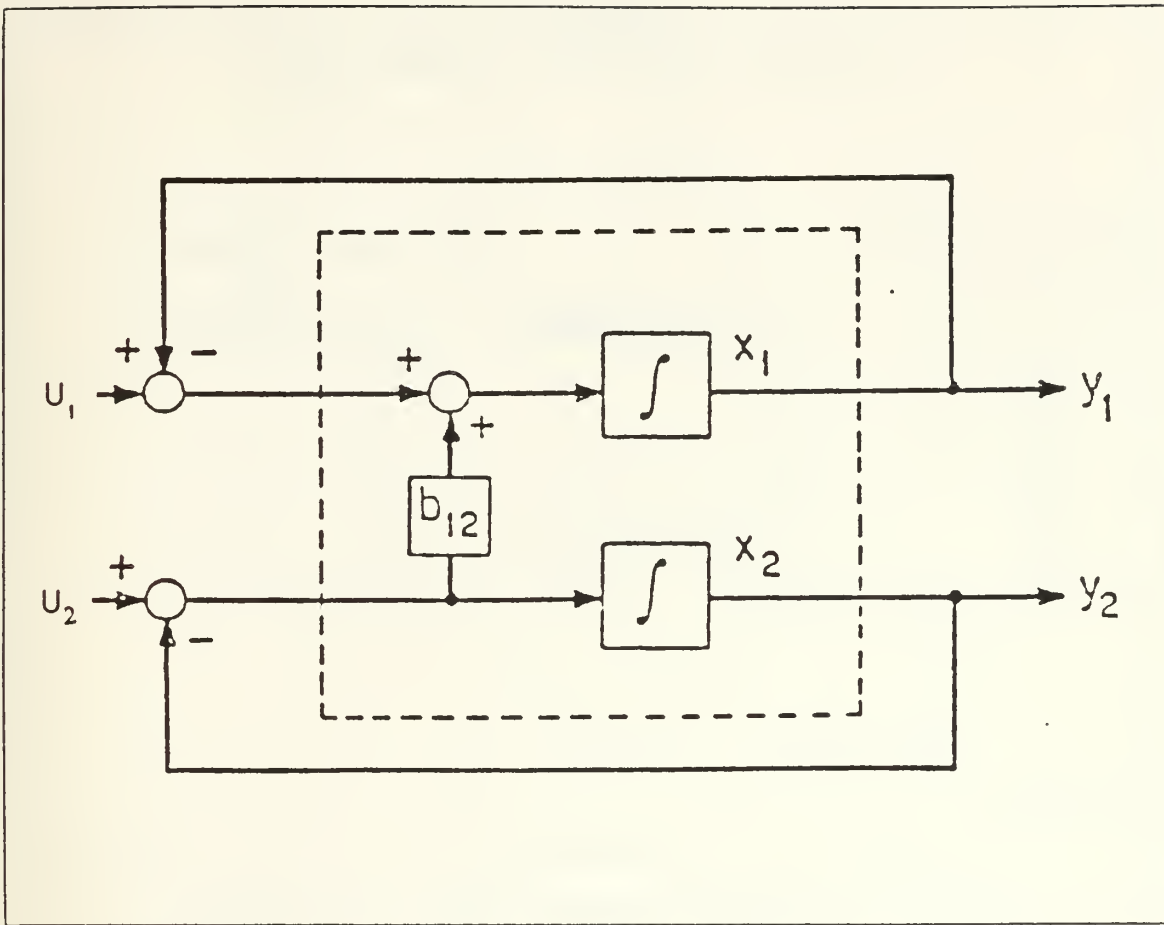


Figure 5.2 Unity Feedback for the 2X2 Model.

system matrix are calculated. The absolute and percentage errors in the eigenvalue due to the perturbation of the closed-loop system are given in Table I .

It can be seen from the above that the first cut design is very susceptible to model and feedback perturbation. A 5 % changes in the f_{21} term has resulted in a large shift (160%) in the closed-loop eigenvalue. Lack of robustness in the unity feedback design can also be seen in terms of the minimum singular value plot of the return difference matrix in the frequency domain. For the unity feedback gain system shown in Figure 5.2, the return difference matrix is given by,

TABLE I
PERTURBED EIGENVALUE FOR 2X2 MODEL

Perturbation in F (+5%)	Absolute Changes in Eigenvalues	Percent Changes in Eigenvalues
f_{11} and f_{22}	0.05 , 0.05	2.5 , 2.5
f_{12}	0.0 , 0.0	0.0 , 0.0
f_{21}	0.67 , +3.265	33.5 , 163.0
All	1,255, 5.346	37.25 , 167.0

$$I + G(s) = \begin{bmatrix} s+2/(s+1) & 50/s+1 \\ 0 & s+2/(s+1) \end{bmatrix}$$

(eqn 5.5)

where $G(s) = C(sI-A)^{-1}B$ is the loop transfer matrix as indicated in Figure 5.3

The multivariable Nyquist diagram (locus of $\det[I+G(s)]$ for the system is shown in Figure 5.4. If this diagram is interpreted as for a single input system, the $(-1/2, \infty)$ gain margin and (± 160) phase margin would lead one to believe that the design is a good one. This has been shown to be not the case, a 5% perturbation in F would cause the system to become unstable. The above clearly demonstrate the inadequacy of the classical method in evaluating stability margin for MIMO system.

Robustness properties of the unity feedback controller will now be analyzed in terms of singular value as discussed

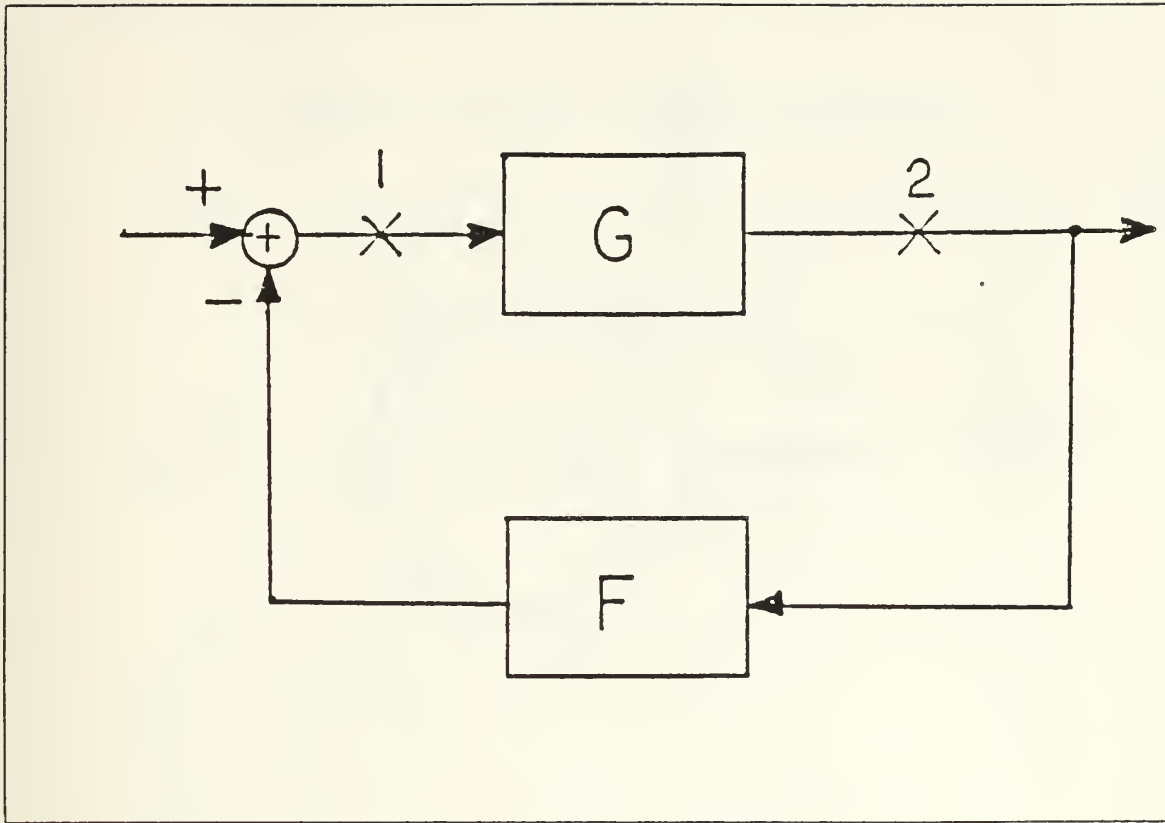


Figure 5.3 Loop Transfer Matrix - $F = I$.

in Chapter 3. The minimum singular value plot of $[I+G(s)]$, is shown in Figure 5.5 as a function of frequency. The lack of robustness is clearly indicated by the relative small singular value at frequency around 2 rad/s. Using the universal phase and gain margin chart developed in [Ref. 19], the minimum singular value at this frequency corresponds to a gain margin of (0.91, 1.0) and a phase margin of (± 4 deg).

It is now shown that formulation using LQ approach and the pole placement procedure developed here will result in robust design that meet the time response requirement. Furthermore, better insight of the design process can be obtained from the procedure to be described here. The first step in the LQ design is to determine the asymptotic

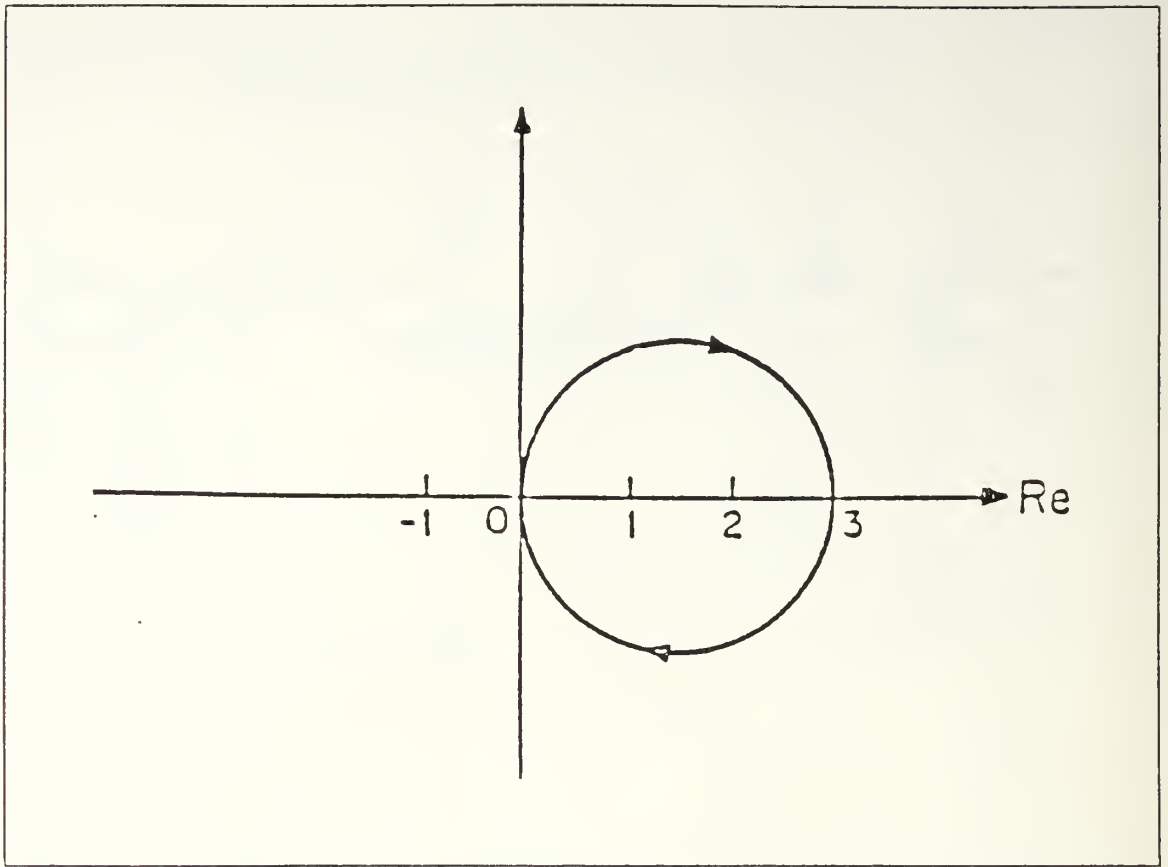


Figure 5.4 Multivariable Nyquist Plot.

properties of the system, i.e. movement of closed-loop poles as $R \rightarrow 0$. Using results from Chapter 2, it can be established that as R increase from 0 to ∞ , both closed-loop poles move from infinity on the real axis to the open-loop poles location. None of the closed-loop poles stay finite as $R \rightarrow 0$ since the dimension of the input control vector is equal to the dimension of the state. Assuming that $R = I$, the pole placement is accomplished in two steps. The first step is to move the open-loop pole at -1 to -2.0. As the system matrix A given is already in Jordan form, no transformation is required. The pole placement program puts the pole at -1.9987 with,

SINGULAR VALUE PLOT

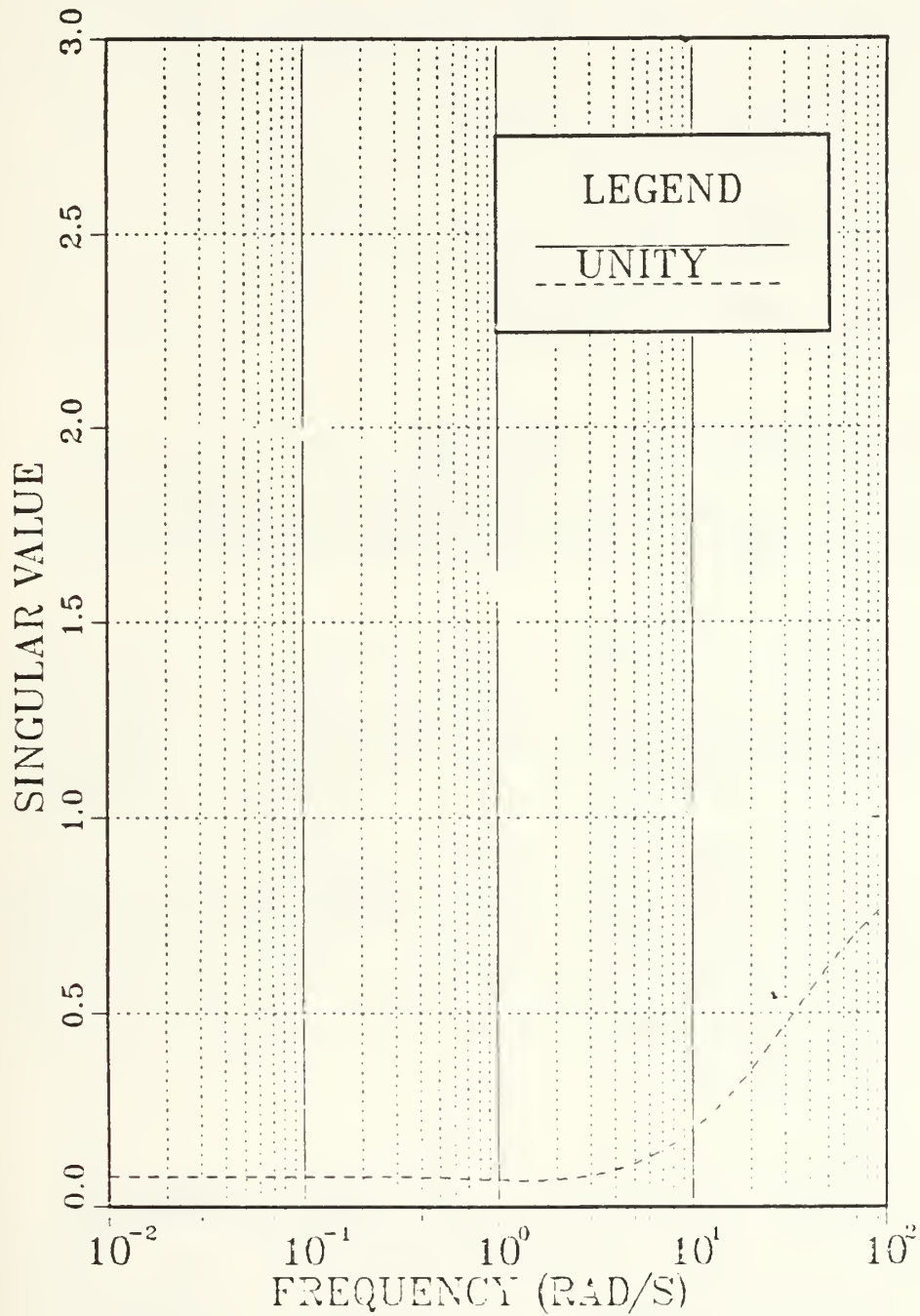


Figure 5.5 Singular Value Plots - Unity Feedback.

$$Q_1 = \begin{bmatrix} 0.00119 & 0 \\ 0 & 0 \end{bmatrix} \quad F_1 = \begin{bmatrix} 0 & 0 \\ 0.01983 & 0 \end{bmatrix}$$

and

$$A_{aug} = \begin{bmatrix} 1.9915 & 0 \\ 0.01983 & 1.0 \end{bmatrix}$$

In the second step, the other open-loop pole at -1. is moved to -2.0 , using A_{aug} as the new plant matrix. The resulting Q and F and the augmented plant matrix become,

$$Q_2 = \begin{bmatrix} -2.998 & -149.9 \\ -149.9 & 7499.3 \end{bmatrix} \quad F_2 = \begin{bmatrix} 0.9994 & -49.978 \\ -0.00841 & 0.4211 \end{bmatrix}$$

The effective Q_e and F_e required to move both open-loop poles at -1.0 to -2. are $Q_e = Q_1 + Q_2$ and $F_e = F_1 + F_2$ as shown in equation 5.6 below. The pole placement procedure is completed with the final eigenvalues placed at $(-1.99255, \pm j0.05628)$.

$$Q_e = \begin{bmatrix} 2.99919 & -149.9 \\ -149.9 & 7499.3 \end{bmatrix} \quad F_e = \begin{bmatrix} 0.9994 & -49.978 \\ 0.0114 & 0.4211 \end{bmatrix}$$

(eqn 5.6)

The singular value plots of the case where $R = I$ together with the unity feedback gain (non-LQ design) are shown in Figure 5.6 The well established fact that LQ design possesses $(1/2, \infty)$ gain margin and $(\pm 60\text{deg})$ phase margin can also be readily observed from the same figure as the minimum singular values of $[I+G]$, $\underline{\sigma}$, is greater than one for all frequency. Changes in closed-loop eigenvalue for a small perturbation in F is again computed as shown in Table II . It can be seen that the LQ design is robust with the largest percentage change in eigenvalues location of only 10%, when compared with the 160% change in the unity feedback design.

TABLE II
PERTURBED EIGENVALUE FOR LQ DESIGN

Perturbation in F (+5%)	Absolute Changes in Eigenvalues	Percent Changes in Eigenvalues
f_{11} and f_{22}	0.2149 , 0.142	10.7 , 7.1
f_{12}	0.119 , 0.119	5.59 , 5.59
f_{21}	0.0592, 0.0578	2.90 , 2.89
All	0.07 , 0.070	3.52 , 3.52

The pole-zero plots of various closed-loop transfer functions of the closed-loop transfer matrix for both the unity feedback and LQ design are compared in Figure 5.7 and 5.8 . For the unity feedback design, zero at minus three for the input 2 to output 1 channel corresponds to the

SINGULAR VALUE PLOT

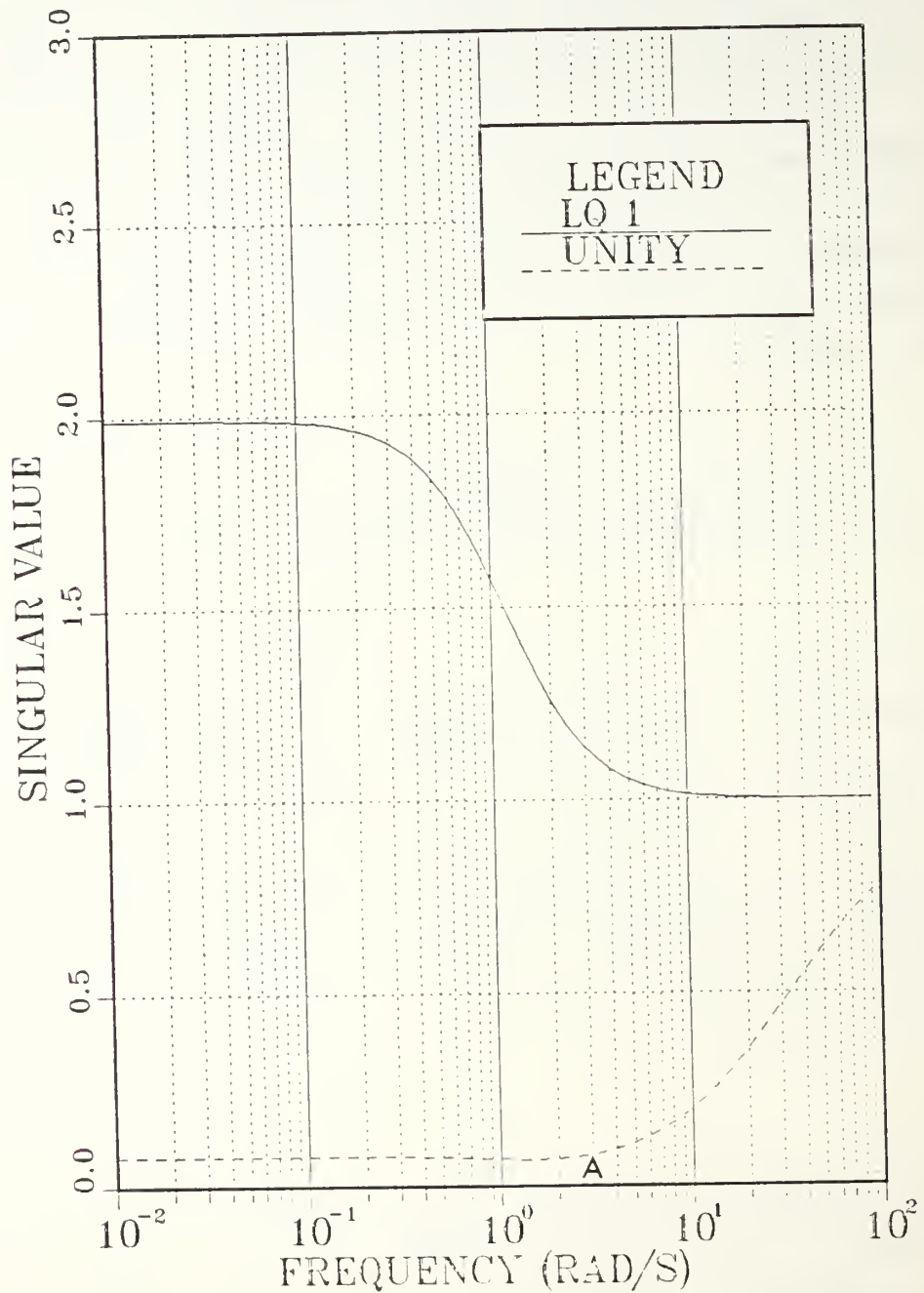
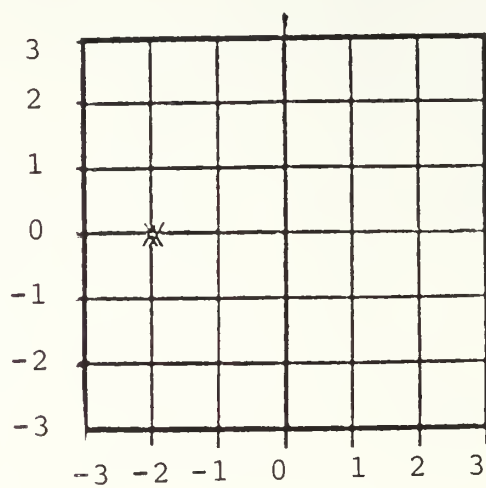


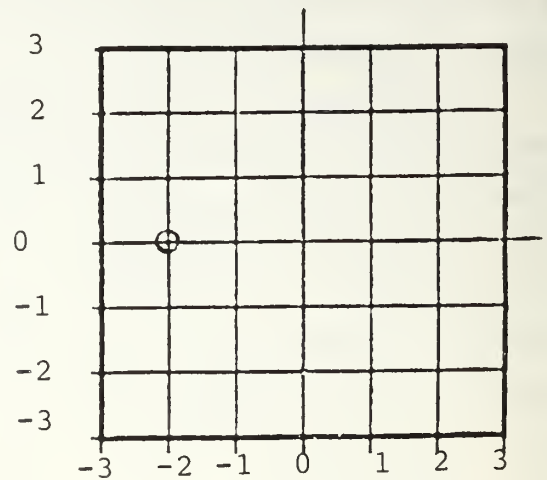
Figure 5.6 Comparison-Singular Value Plots (2x2 model).

minimum singular value frequency (point A in Figure 5.6). It is in fact this zero that causes the the system to be sensitive to perturbation. For the LQ design, the built in robustness cause the zero at minus three to move toward the pole location at -2.

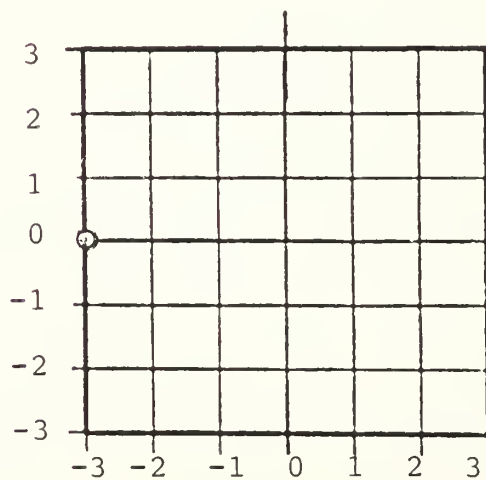
Improvement in the robustness from LQ design is now analyzed in terms of Bode plots. The open-loop gain and phase vs frequency plots for a MIMO system can be obtained from the open-loop transfer function matrix, $G(s)$. For full state feedback, $G(s)=F(sI-A)^{-1}B$. (if the open-loop plots without feedback is considered, $G(s)=C(sI-A)^{-1}B$) The matrix G is a matrix-valued rational function of s , it describes how the system (with or w/o feedback) appears to its environment. It is an external description of the system and is closely related to the zeros of the system. The open-loop Bode plots for the two designs are compared in Figures 5.9 to 5.11 As $b_{21}=0$, there is no coupling from channel 1 to channel 2. All channels have a -20 dB/decade slope at high frequency which is in agreement with earlier observation that each channels has one finite zero. It is interesting to note that the unity feedback and LQ design result in almost identical gain vs frequency plots for direct channels (i.e. channel 1-1 and 2-2). Any classical single loop type of analysis will not be able to detect any difference between the two designs. On the other hand, cross-coupling effect can be readily seen from the gain vs frequency plot in channel 2-1 (Figure 5.10). The unity feedback design is characterized by the rather large channel 2-1 gain at low frequency. A very small perturbation in channel one's parameter can change the system behavior considerably. This has been illustrated earlier by perturbing the feedback gain. It can be seen from the figure that LQ design reduces the crossfeed gain by large amount.



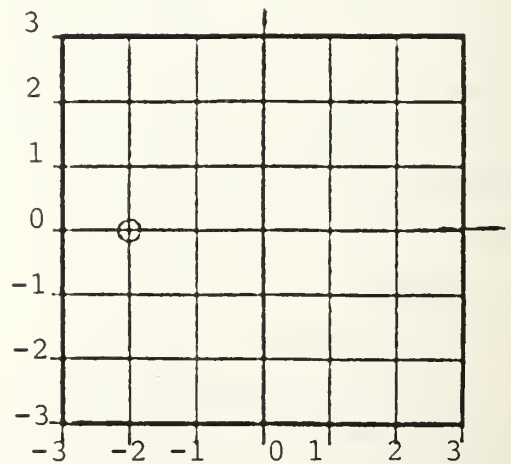
POLES (ALL)



ZERO (1 TO 1)

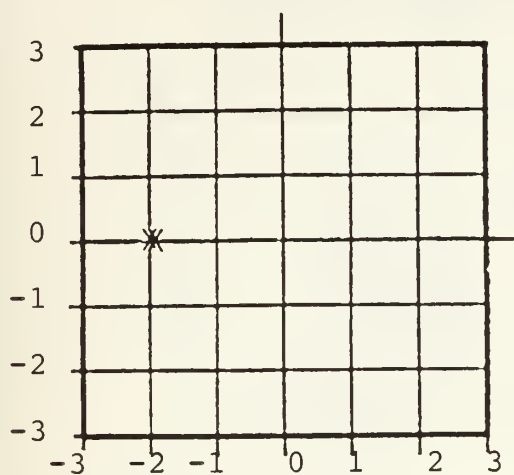


ZERO (2 TO 1)

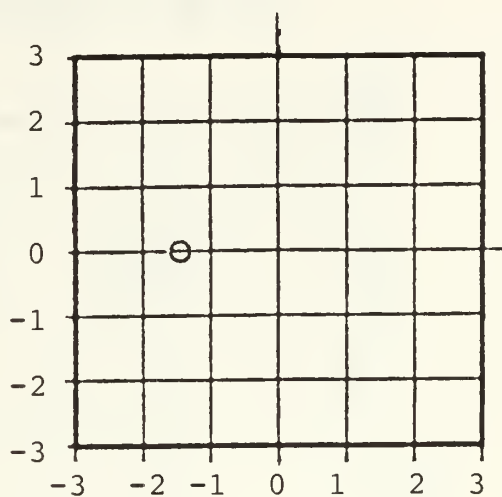


ZERO (2 TO 2)

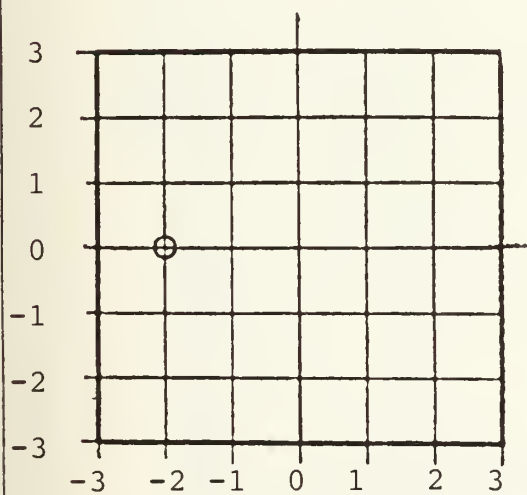
Figure 5.7 Closed-Loop Pole-Zero Plots (Unity Gain FB).



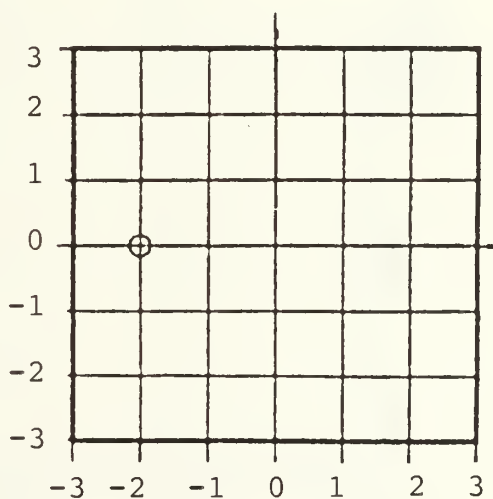
POLES (ALL)



ZERO (1 TO 1)



ZERO (2 TO 1)



ZERO (2 TO 2)

Figure 5.8 Closed-Loop Pole-Zero Plots (LQ Design).

OPEN LOOP GAIN 1 - 1

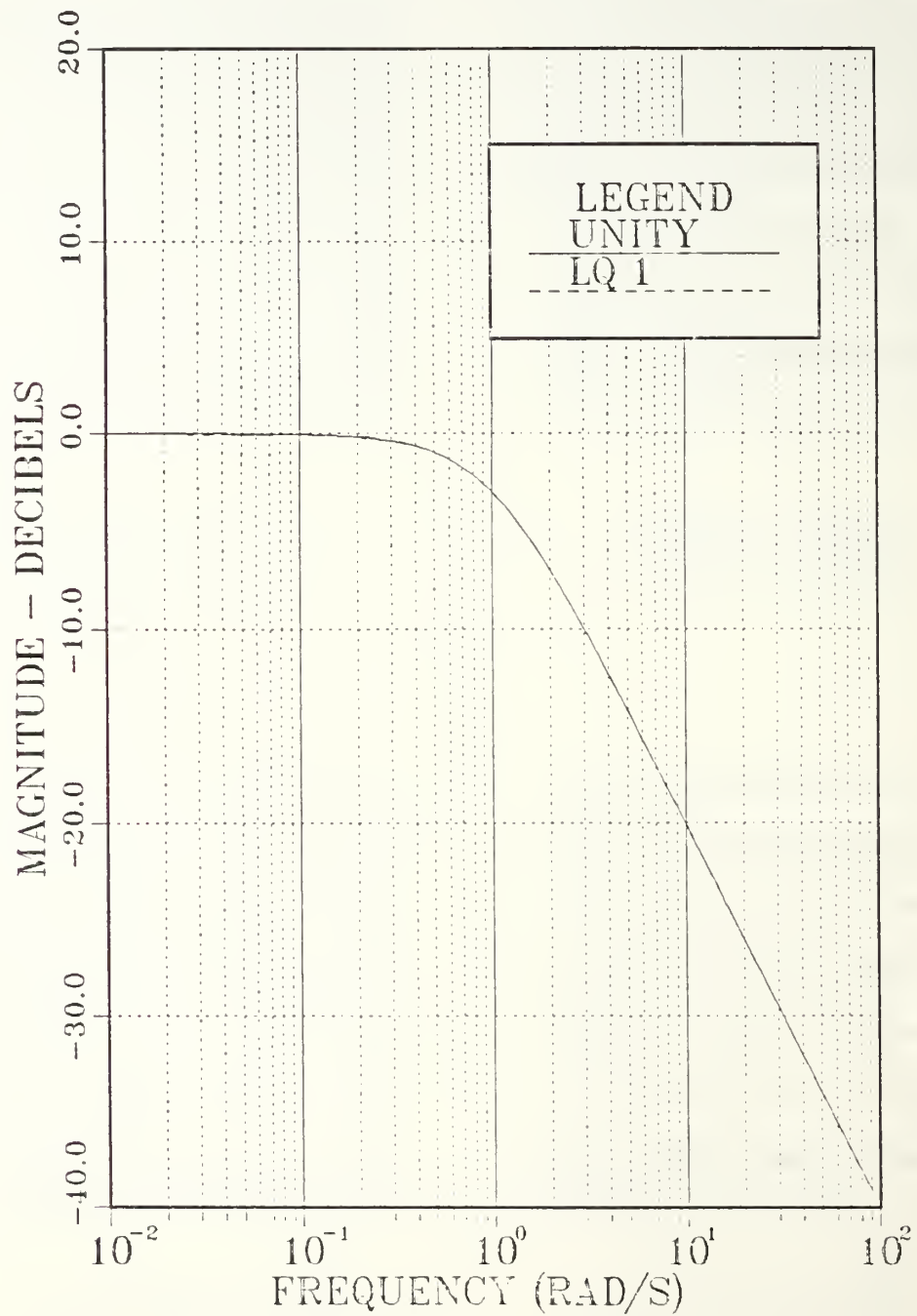


Figure 5.9 Open-Loop Bode Plots Channel 1-1.

OPEN LOOP GAIN 2 - 1

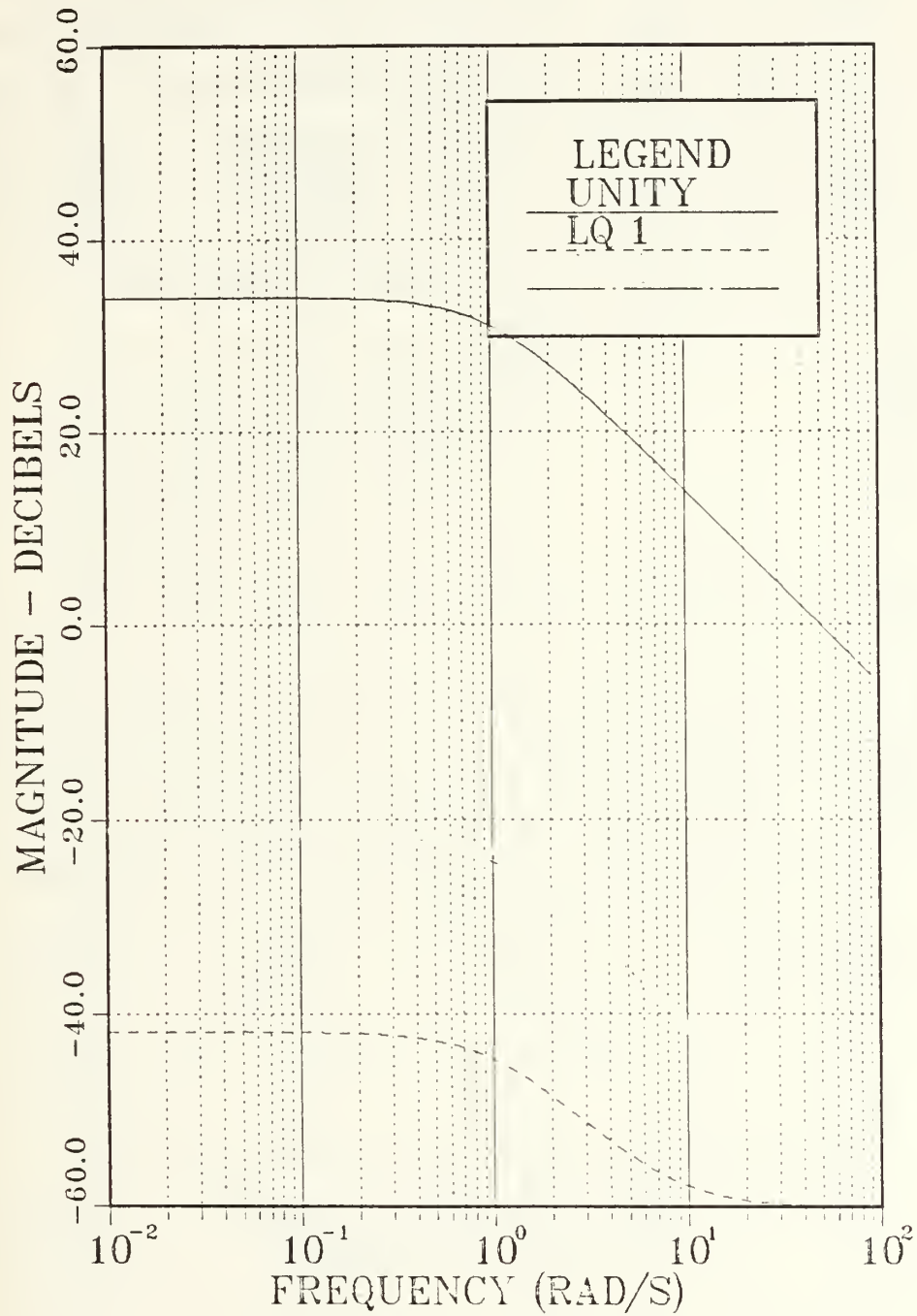


Figure 5.10 Open-Loop Bode Plots Channel 2-1.

OPEN LOOP GAIN 2 - 2

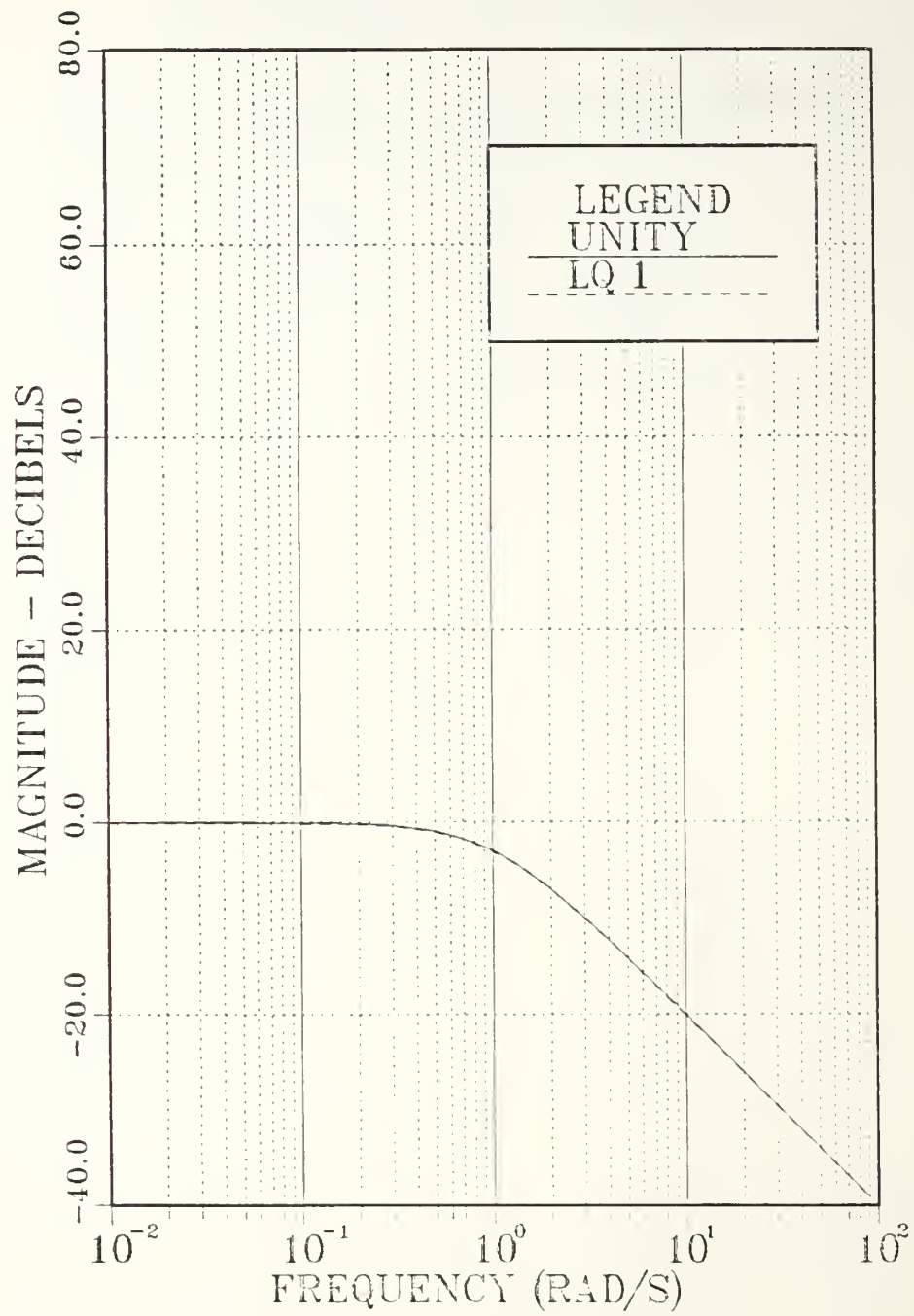


Figure 5.11 Open-Loop Bode Plots Channel 2-2.

B. A HELICOPTER DESIGN PROBLEM

The design procedure described in Chapter 4 is further illustrated in this section by an actual design problem. A controller is designed using the Linear Quadratic Pole Placement approach for the lateral dynamic model of a CH-47 helicopter. The resulting design is then compared to multi-variable state feedback controller given in [Ref. 23].

The highly coupled two inputs lateral axis model of the CH-47 helicopter is used as a full order system. The state vector $x(t)$ and control input vector $u(t)$ are given by,

$$x_1 = v = \text{Y-axis velocity (ft/sec)}$$

$$x_2 = p = \text{Roll rate (rad/sec)}$$

$$x_3 = r = \text{Yaw rate (rad/sec)}$$

$$x_4 = \phi = \text{Roll angle (rad)}$$

$$u_1 = \delta_b = \text{Yaw rate rotor deflection control (inches.)}$$

$$u_2 = \delta_c = \text{Roll rate rotor deflection control (inches.)}$$

The state variables and the body axes of the aircraft are illustrated in Figure 5.12. The yaw and roll rotor deflection control produce changes in the yaw rate, side slip angle, roll rate and bank angle. Assuming full state feedback, the A, B, and C system matrices are given by,

$$A = \begin{bmatrix} -2.27 & 1.420 & -0.15 & 31.99 \\ .01 & -0.7 & -0.07 & 0.0 \\ 0.04 & -0.05 & -0.05 & 0. \\ 0. & 1. & 0.11 & 0. \end{bmatrix}$$

$$B = \begin{bmatrix} 0.12 & 0.95 \\ 0.04 & -8.37 \\ .34 & .020 \\ 0.0 & .0 \end{bmatrix}$$

$$C = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

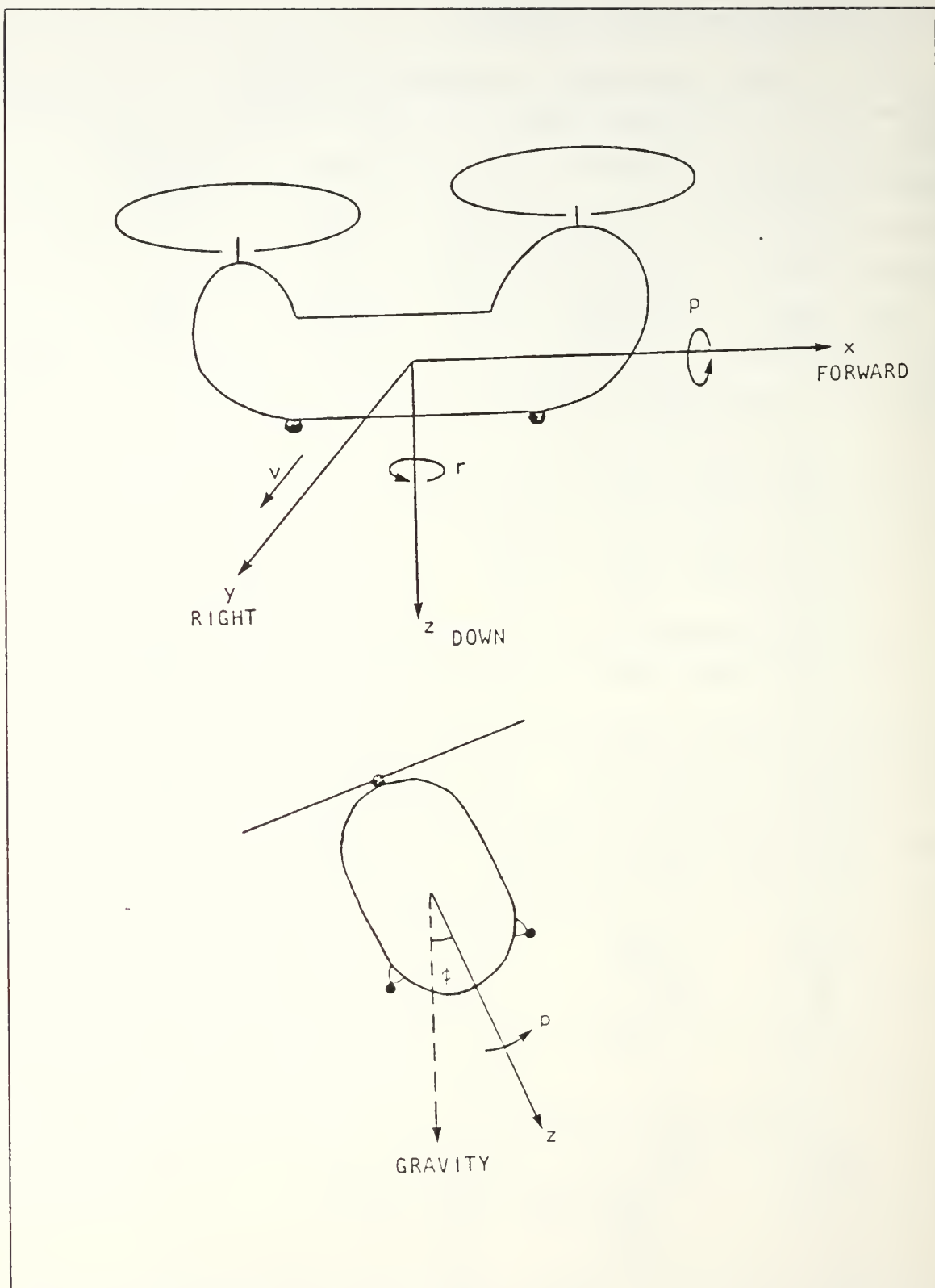


Figure 5.12 State Variables and Body Axes for CH-47.

The open loop eigenvalues of this system are;

$$\begin{aligned}s_1 &= +0.2065 \\s_2 &= -0.0503 \\s_3 &= -1.0498 \\s_4 &= -2.1263\end{aligned}$$

The open-loop system is not stable and the time response is shown in Figure 5.13 for zero input and an initial condition of $\phi(0)=0.1$ rad

The design requirements are to satisfy specification in terms of the step input response of the roll attitude channel (ϕ/ϕ_c). Stability margin requirements are those given by standard military specification. In [Ref. 23] three designs were obtained to satisfy the desired time response performance specification. All three control laws are of the form given by,

$$u(t) = -Fx(t) + h\phi_c(t) \quad (\text{eqn 5.7})$$

The values of F and h are summarised in Table III. It was shown in [Ref. 23] that two of the designs (design 1 and 2) were extremely sensitive to model errors and perturbations. It is now shown that LQ formulation using the pole placement procedure developed here will result in robustness design and yet satisfy the conventional time response criteria.

It is assumed that the closed-loop poles requirement are the same as those obtained in the AlphaTech's design 1. $(-25.12, -12.51, -9.652, -2.125)$. The first step in the design is to establish some asymptotic properties of the system. The dimension of the control input (2) is less than the dimension of the state(4). As shown in Chapter 2, this implies that at least two closed-loop poles approach minus infinity in the complex plane when $R \rightarrow 0$. When $R \rightarrow 0$ (meaning no control input allowed), the closed-loop poles

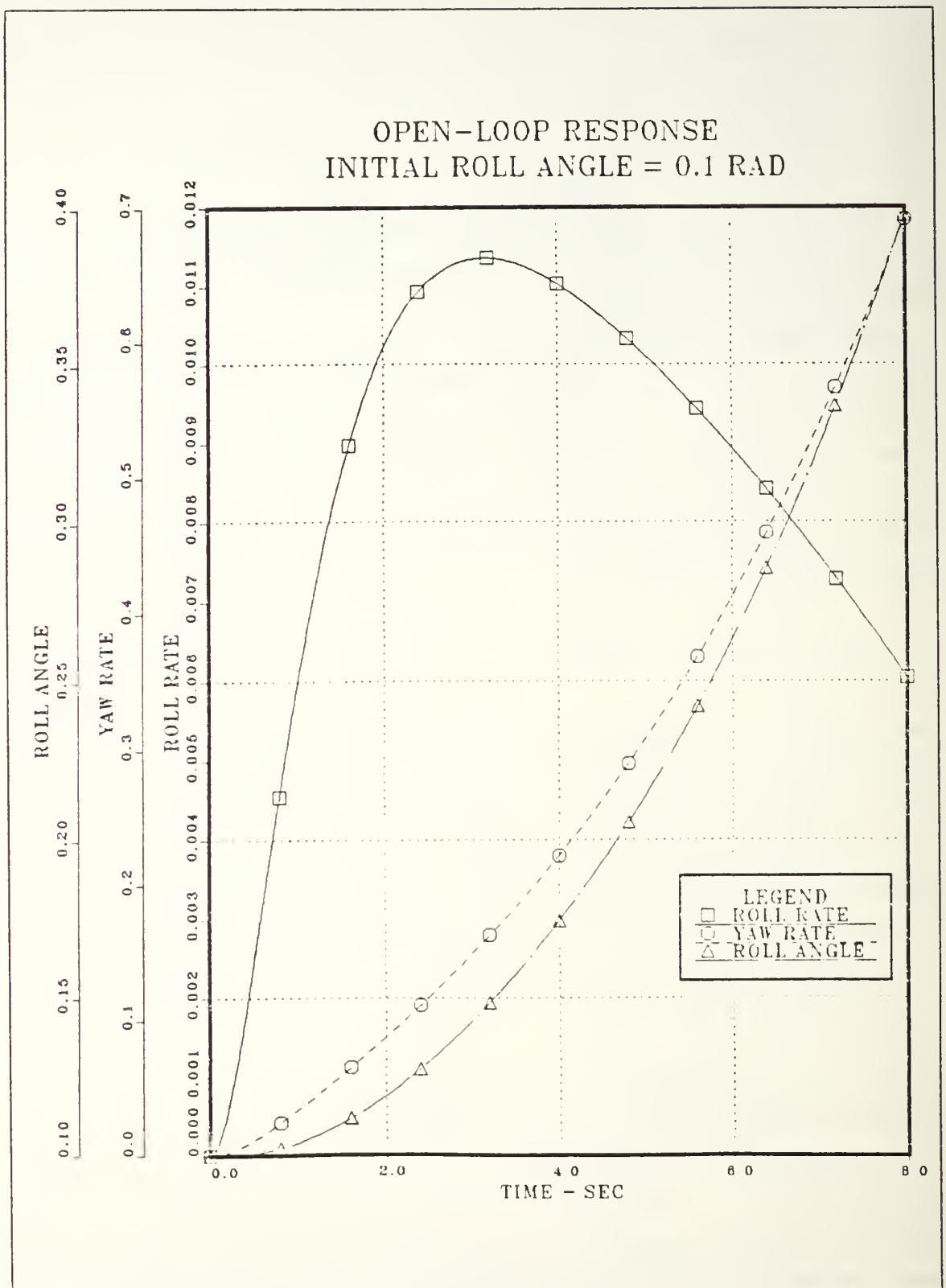


Figure 5.13 Open Loop Time Response $\phi(0) = 0.1$ rad.

TABLE III
ALPHATECH DESIGNS (1, 2, AND 3)

Designs	Feedback Gains (F)				h
one	-1.72 0.024	-23.5 -2.71	70.6 0.368	595 -7.99	595 -7.99
two	0.198 -0.01	154. -1.592	18.3 -0.189	142.0 -1.47	142.0 -1.47
three	0 0	0 -4	25.5 0	0 -27	0 -27

approach the open-loop poles or their mirror images if they are in the right hand plane. (i.e. -0.2065, -0.0503, -1.04987, and -2.12635).

The next step in the design procedure is to select a suitable starting control weighting matrix. Based on guideline given in [Ref. 4] and assuming that u_{1max} and u_{2max} are equal to 1 inch. $R = I$ is selected for the initial design. (In the present formulation ρ is not needed as only Q is varied to place the pole).

The sequence of the reassignment is determined next. At present there is no known established guideline for selecting the preferred sequence of moving the poles. Two extreme sequences are considered as follows,

A. Move the left most open-loop pole to the left most desired closed-loop pole etc. The reassignment sequence then become:

1. move pole at -2.1265 to -25.12
2. move pole at -1.0483 to -12.51

3. move pole at -0.2065 to -9.652
4. move pole at -0.0503 to -2.125

B. Move the right most open-loop pole to the left most desired closed-loop pole etc. For the problem given, the reassignment sequence are :

1. move pole at +0.2065 to -25.12
 2. move pole at -0.0503 to -12.51
 3. move pole at -1.0498 to -9.652
 4. move pole at -2.126 to -2.125
- (close, so no move required)

Using the pole placement program developed in this thesis, the corresponding Q and F for each of the reassignments are obtained and tabulated in Tables IV and V for the two cases selected (design LQ-A and LQ-B). It can be seen from the table that difference reassignment sequence results in different set of Q_e and F_e . In general, once the matrix R and the n closed-loop poles are selected, there are $n(n+1)/2$ extra degrees of freedom available in Q . This also verifies the well-known fact that the general solutions of Q and R for a given set of closed-loop eigenvalue are non-unique. The elements of the Q matrix obtained during each reassignment depends on the transformation matrix (M , L or U) and hence the eigenvectors that are used to construct them. The important of eigenvector type of assignment is evident as Q , F and the resulting closed-loop poles depend on the eigenvectors used. The designer can shape the design by choosing appropriate eigenvectors for the transformation matrix. The application of these extra degrees of freedom will be discussed in the following section. The resulting designs (LQ-A and LQ-B) are now compared with the first design in [Ref. 23].

TABLE IV
RESULTS FROM POLE PLACEMENT SEQUENCE(LQ-A)

Move	Q and F obtained during each reassignment			
Q ₁	0.10027	0.96194	0.11959	-1.50849
	0.96194	9.22847	1.14725	-14.47182
F ₁	0.11959	1.14725	0.14262	-1.79908
	-1.50849	-14.47182	-1.79908	22.69429
Q ₂	0.00387	0.03151	0.00399	-0.06416
	-0.24260	-2.83522	-0.34541	3.11642
Q ₃	6.12853	-2.99198	0.51376	-95.14285
	9198	1.46070	-0.25082	46.44916
F ₂	0.51376	-0.25082	0.04307	-7.97589
	-95.14284	46.44916	-7.97589	1477.05225
Q ₄	0.05825	-0.02844	0.00490	-0.90426
	2.27529	-1.11079	0.19079	-35.32260
Q ₅	80.37563	18.91502	13.64762	218.27522
	18.91502	4.45133	3.21173	51.36732
F ₃	13.64761	3.21173	2.31734	37.06267
	218.27521	51.36732	37.06267	592.76758
Q ₆	1.58772	0.37365	0.26961	4.31182
	-8.62241	-2.02917	-1.46414	-23.41618
Q ₇	0.15575	0.02839	-2.48989	0.19453
	0.02839	0.00517	-0.45383	0.03546
F ₄	-2.48989	-0.45383	39.80322	-3.10981
	.19453	0.03546	-3.10981	0.24297
Q ₈	-0.38007	-0.06927	6.07574	-0.47469
	-0.06411	-0.01168	1.02482	-0.08007
Q ₉	86.76016	16.91336	11.79108	121.81841
	16.91336	15.14567	3.65433	83.38010
F ₅	11.79107	3.65433	42.30624	24.17787
	121.81841	83.38010	24.17787	2092.75684
Q ₁₀	1.26977	0.30745	6.35424	2.86871
	-6.65383	-5.98686	-0.59394	-55.70242

$$u(t) = -Fx(t) + h\phi_c(t), \quad h = \begin{bmatrix} 2.8687 \\ -55.7024 \end{bmatrix}$$

TABLE V
RESULTS FROM POLE PLACEMENT SEQUENCE(LQ-B)

Move	Q and F obtained during each reassignment			
Q ₁	0.00031	0.05289	0.00589	0.04781
	0.05289	9.06491	1.00865	8.19296
	0.00589	1.00865	0.11223	0.91163
	0.04781	8.19296	0.91163	7.40489
F ₁	0.00017	0.02851	0.00317	0.02576
	-0.01771	-3.03548	-0.33776	-2.74347
Q ₂	0.26982	0.46312	16.33746	10.89106
	0.46312	0.79492	28.04224	18.69380
	16.33746	28.04222	989.23975	659.45752
	10.89106	18.69380	659.45728	439.61426
F ₂	0.44607	0.76563	27.00978	18.00517
	-0.26207	-0.44982	-15.86824	-10.57837
Q ₃	0.16317	-0.35205	6.24519	-9.35123
	-0.35205	0.75956	-13.47424	20.17563
	6.24519	-13.47425	239.02527	-357.90405
	-9.35123	20.17563	-357.90381	535.90698
F ₃	0.19939	-0.43018	7.63129	-11.42668
	0.30220	-0.65201	11.56628	-17.31876
Q _e	0.43330	0.16396	22.58853	1.58764
	0.16396	10.61939	15.57663	47.06238
	22.58853	15.57661	1228.37695	302.46509
	1.58764	47.06238	302.46509	982.92603
F _e	0.64563	0.36396	34.64423	6.60425
	0.02242	-4.13731	-4.63971	-30.64059

$$u(t) = -F_x(t) + h \phi_c(t), \quad h = \begin{bmatrix} 6.60425 \\ -30.64059 \end{bmatrix}$$

Closed loop time response for a step input in roll attitude command ($\phi_c = 0.1$ rad) for the three designs are shown in Figures 5.14 to 5.16. It can be seen from the response plots that although all three designs have almost identical closed-loop pole location, the step response for various states differs. The AlphaTech design's response is characterized by the large 'overshoot' in the yaw rate step response. In the Linear Quadratic Design (LQ-A), the 'overshoot' occurs with the roll attitude response rather than

the yaw rate response; the yaw rate response is well damped. The LQ-B design appears to be a better design as coupling among modes are small and can not be detected from the time response plot. It is noted that the difference in response for the same set of closed-loop eigenvalue is due to coupling among various modes through their respective eigenvectors. The 'overshoot' in this case is obviously not due to complex conjugate poles as all closed-loop poles in the three designs are on the real axis. The set of final eigenvector for the three designs are tabulated in Table VI. Inter-mode coupling for the yaw rate response in the AlphaTech design and roll attitude response in the LQ design case A can be readily seen from the table. The issue of eigenvector assignment will be discussed in the next section.

TABLE VI
CLOSED-LOOP EIGENVECTORS

Eigenvalues/Eigenvector				
Alpha one	-24.79907	-2.12242	-9.72403	-11.9449
	-0.39336	-0.99976	-0.48016	-0.41854
	0.10767	0.00822	-0.32948	-0.33640
	-0.91306	0.01955	-0.81181	-0.84283
	-0.00029	-0.00488	0.04306	0.03593
LQ-A	-2.12499	-9.26645	-12.52066	-25.20990
	0.22870	-0.15459	0.01894	-0.12653
	0.18018	-0.98218	-0.99664	0.99118
	-0.95603	-0.01258	-0.00396	0.00031
	-0.03529	0.10614	0.07964	-0.03932
LQ-B	-25.17894	-9.62708	-12.51644	-2.12630
	-0.12652	-0.13209	0.05042	-0.99977
	0.99118	-0.98557	-0.99375	0.00797
	0.00024	-0.02568	0.06075	0.01945
	-0.03937	0.10267	0.07886	-0.00475

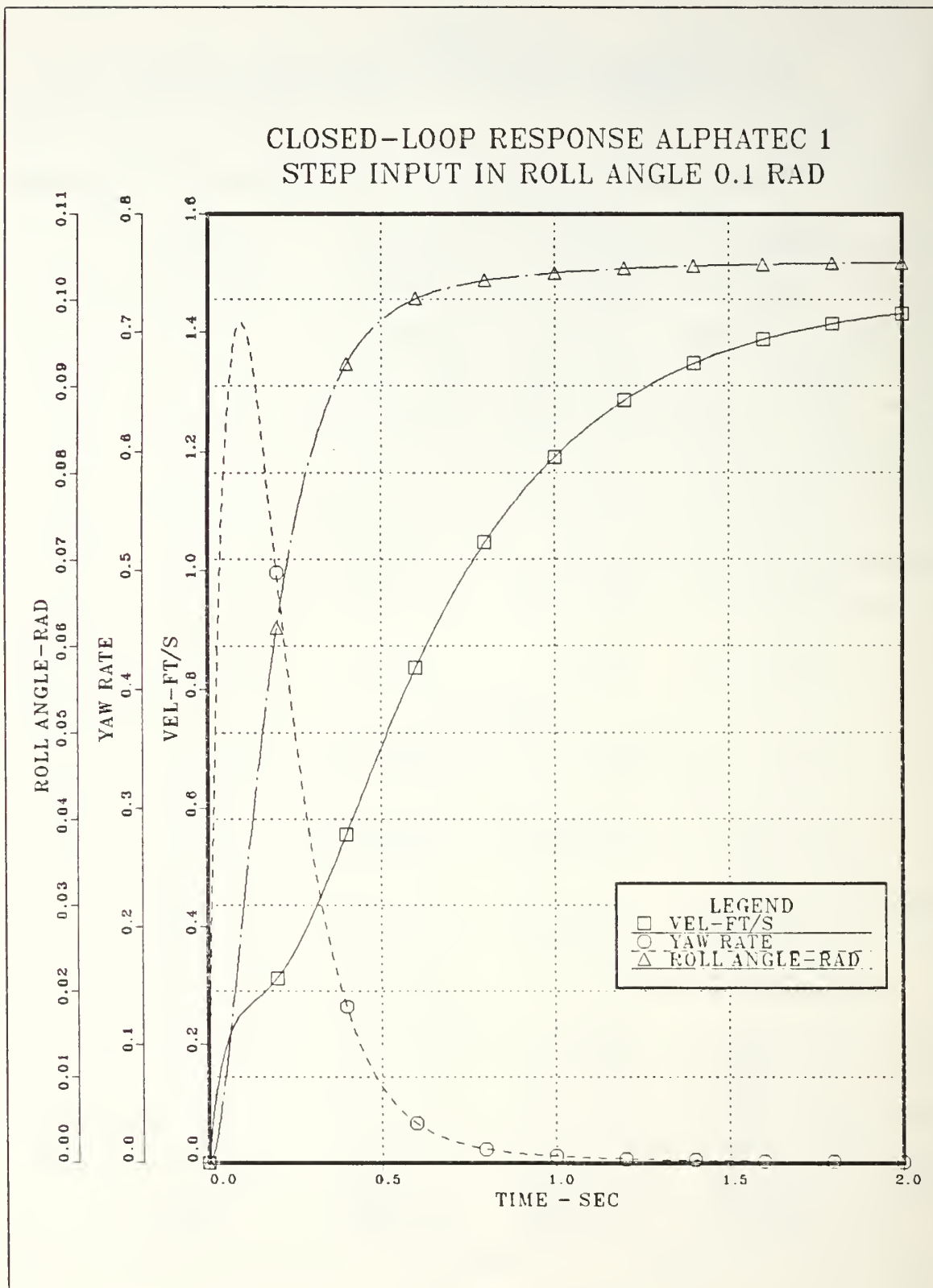


Figure 5.14 Closed-Loop Time Response Plot (AlphaTech 1).

CLOSED LOOP RESPONSE LQ-A STEP INPUT IN ROLL ANGLE 0.1 RAD

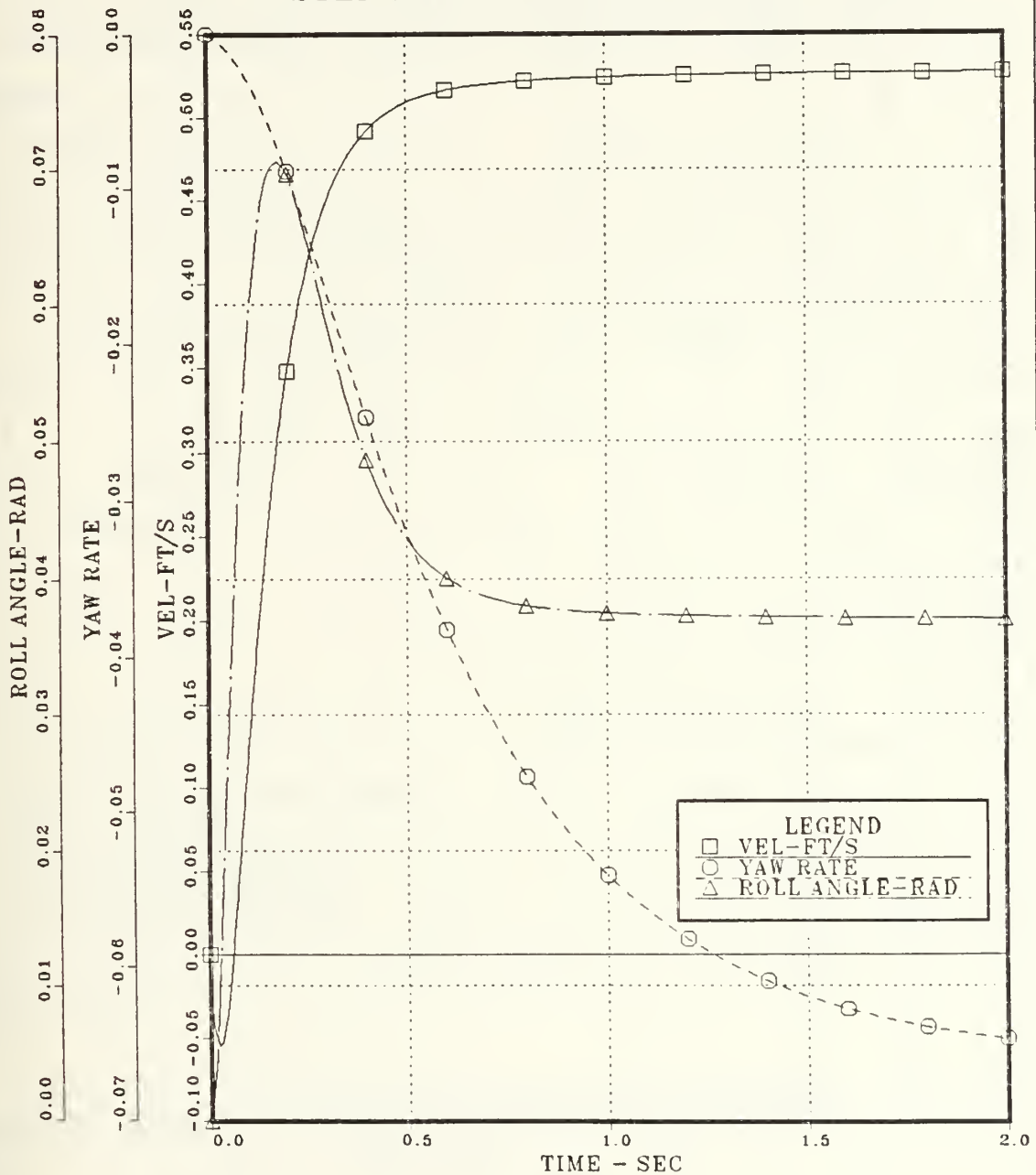


Figure 5.15 Closed-Loop Time Response Plot (LQ-A).

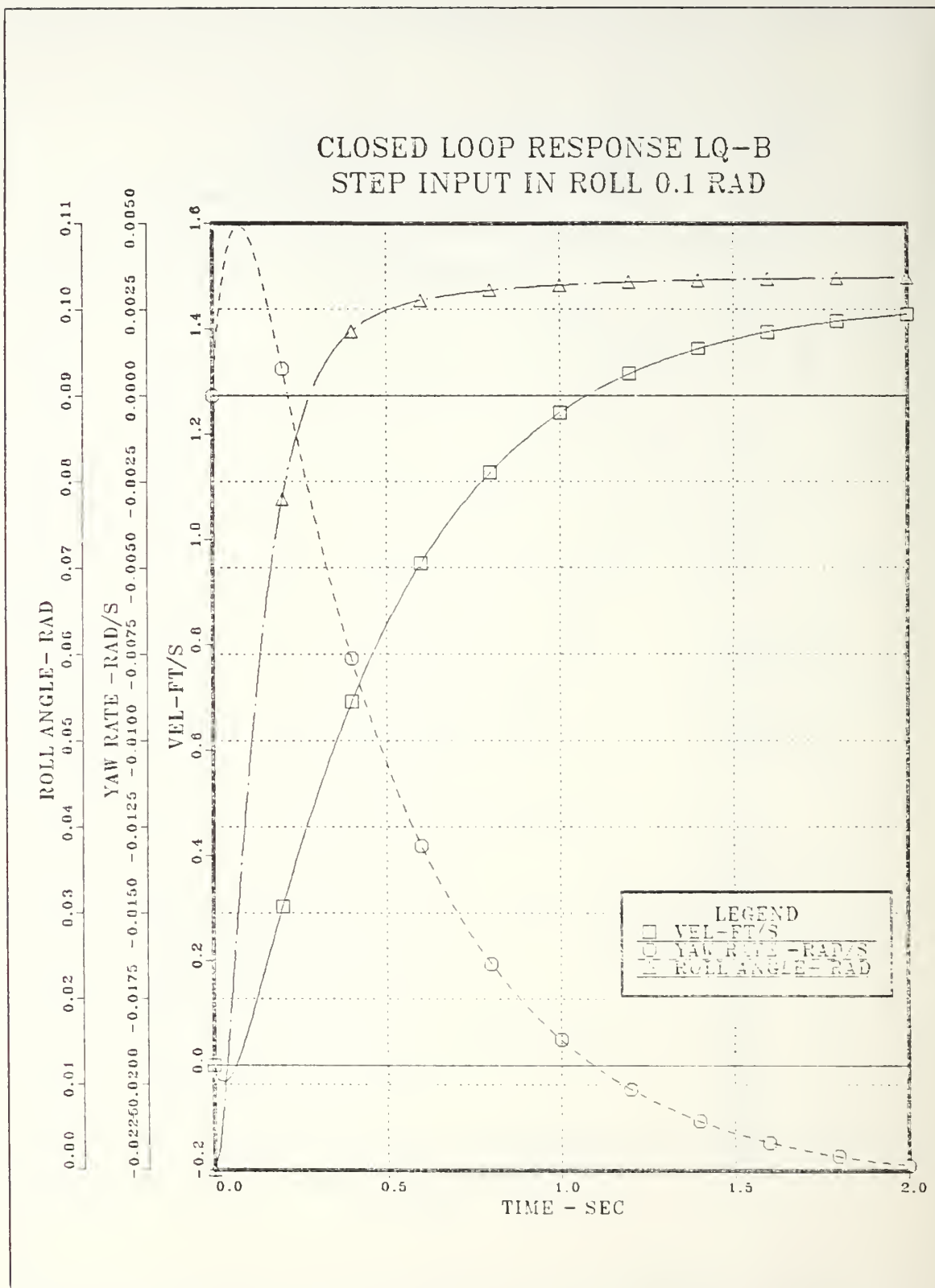
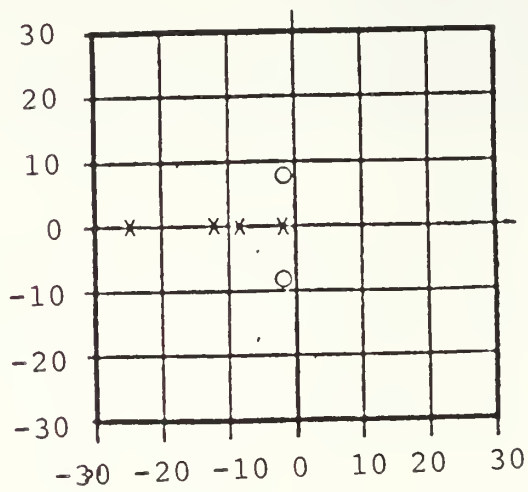


Figure 5.16 Closed-Loop Time Response Plot (LQ-B).

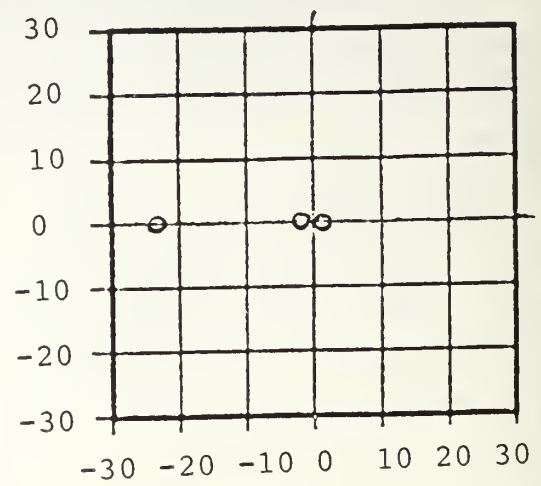
The difference in time response performance for the three designs can also be interpreted in term of the closed-loop pole-zero plots in the frequency domain. The pole-zero plots for the designs, with ϕ_c as the input, are shown in Figures 5.17 to 5.19 . As all pole locations are the same for the three designs, they are indicated only once in each diagram.

In the AlphaTech's design, the zero locations for the various channels clearly indicate the weakness in the design. Most transmissions zeros are located away from the poles locations. The system is therefore strongly coupled to its external environment. A good example is the lightly "damped" pair of zero at $(-1.05, \pm 7.31j)$ for the $v-\phi_c$ channels. It is in fact these undesired zeros that reduce the overall robustness of the system. The mechanism of robustness improvement in the LQ design can also be seen in the pole-zero plots in Figures 5.18 and 5.19 The built-in robustness in the LQ design causes the zeros at various channels to move to locations where their transmission properties can be canceled by the closed-loop poles. An excellent example is in the $v-\phi_c$ channel, where the zeros at -2, -7, and the mirror image of +24.0 are fairly close to the closed-loop pole locations at -2.12, -9.26 and -25.12. In the case where the zero from LQ design are not close to the closed-loop poles ($r-\phi_c$ channel for LQ-A) the zeros are "well damped" and their transmission properties can be neglected.

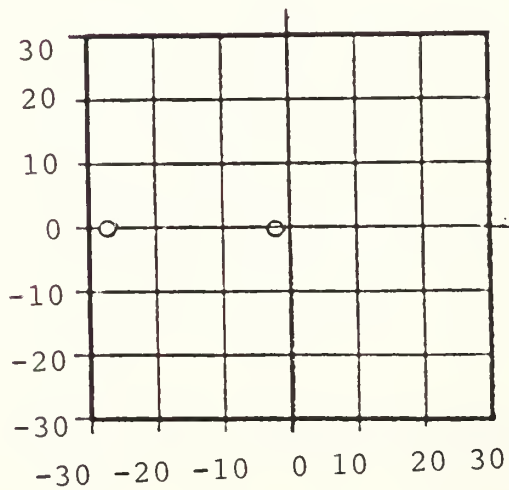
Robustness properties of the three designs presented here can also be analyzed from the open-loop Bode plots. The open-loop transfer function gain plots for channel 1-1, 1-2, 2-1 and 2-2 for the three designs are shown in Figures 5.20 to 5.24 Cross coupling problem for the AlphaTech design is clearly indicated by the relatively high gain of



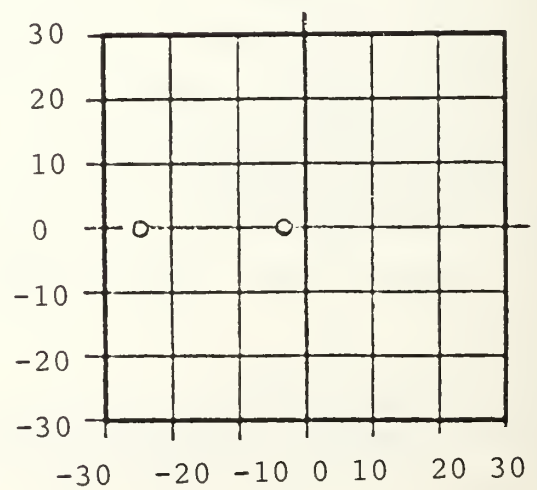
$$V - \phi_c$$



$$\dot{\phi} - \phi_c$$

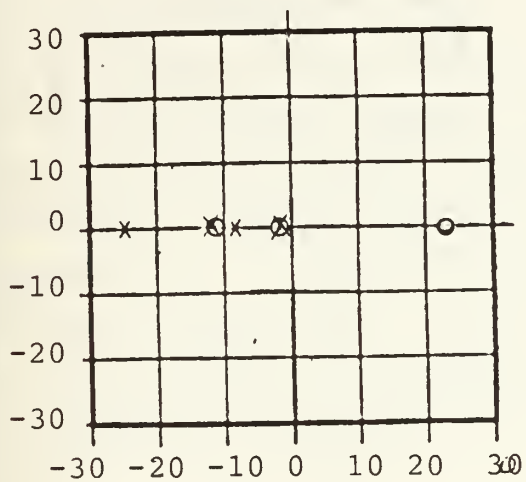


$$R - \phi_c$$

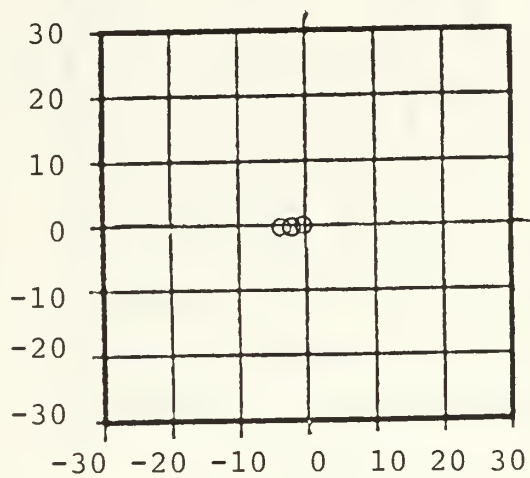


$$\phi - \phi_c$$

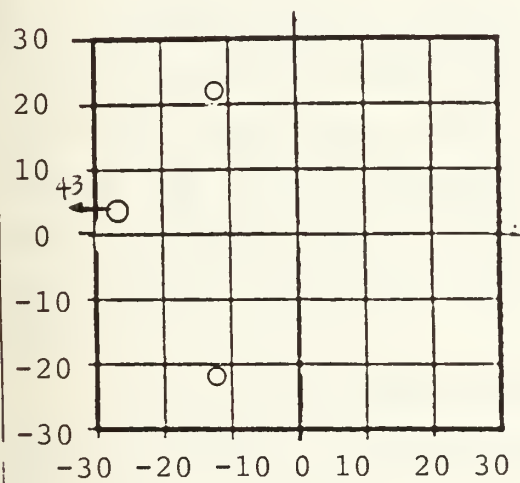
Figure 5.17 Closed-Loop Pole-Zero Plots (AlphaTech 1).



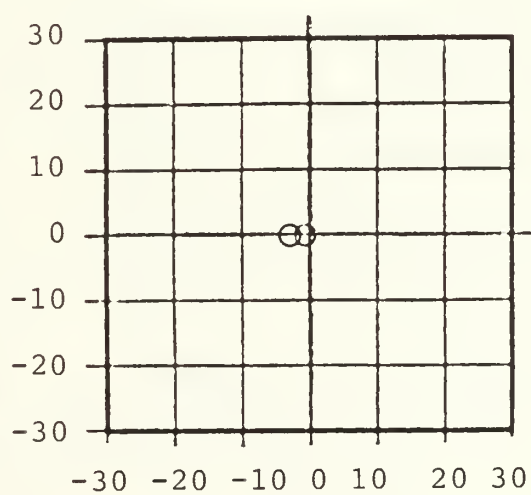
$$V - \phi_c$$



$$\dot{\phi} - \phi_c$$

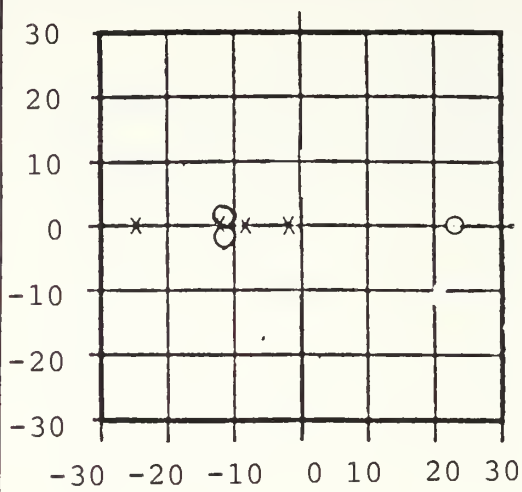


$$R - \phi_c$$

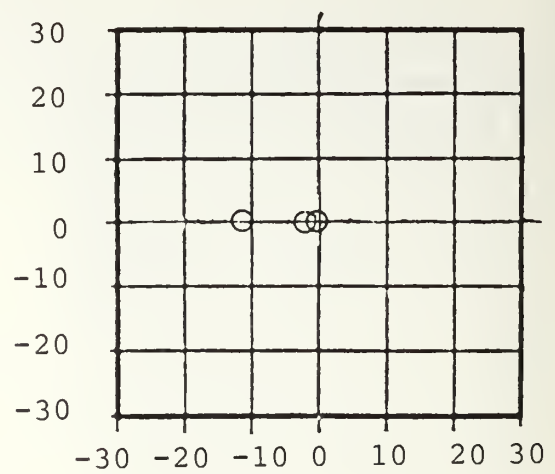


$$\phi - \phi_c$$

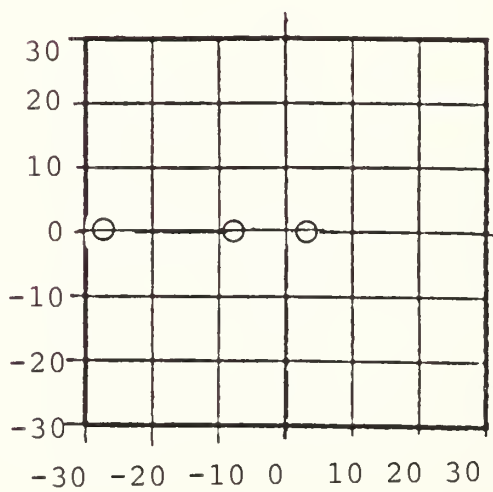
Figure 5.18 Closed-Loop Pole-Zero Plots (LQ-A).



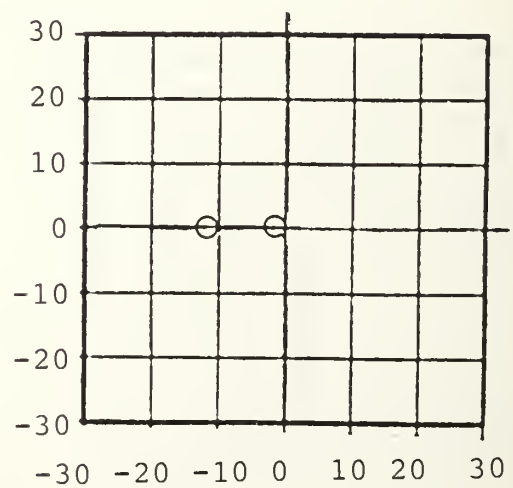
$$V - \phi_c$$



$$\dot{\phi} - \phi_c$$



$$R - \phi_c$$



$$\phi - \phi_c$$

Figure 5.19 Closed-Loop Pole-Zero Plots (LQ-B).

the channel from input 2 to input 1 (2-1). Both LQ designs presented here reduced the gain in this channel by more than 40 dB in the frequencies of interest. The bandwidth is reduced from above 100 rad/s to about 8 rad/s. Unlike the simple (2x2) system where there was little difference between the direct channels of the two designs, gain adjustment is observed in all channels. For example, LQ designs reduce the gain in channel 1-1 but increase the gain in channel 2-2. There is also a slight increase in the coupling channel 1-2. The overall effect is that of gain balancing, gains in channels that are affected by cross-coupling perturbation are lowered together with some adjustments in other channels. Different reassignment sequence results in different adjustments. The designer has to choose a set of gain curve depending on the particular requirement. The relationship between the open-loop Bode plots and the zeros of the system is also evident from these diagrams. The low frequency resonance (or 'peak') in the gain vs frequency plot for the AlphaTech design correspond to the undesirable zero mentioned earlier. In the LQ designs, these low frequency resonances are absent because of the more desirable zero locations. It is also interesting to note that the LQ-B gain curves fall nicely in between the other two designs. This fact, together with its better overall low frequency gain characteristic, may account for the better time response behavior of the LQ-B design.

As a final comparison, the closed-loop frequency response plots of the various channel for the three designs are shown in Figures 5.25 to 5.32. At high frequency, all the gain plots approach the -20dB/decade slope as all eigenvalues are on the real axis. The AlphaTech design has a near 0db gain for frequency up to about 2 rad/s. It appears

OPEN LOOP GAIN 1 - 1

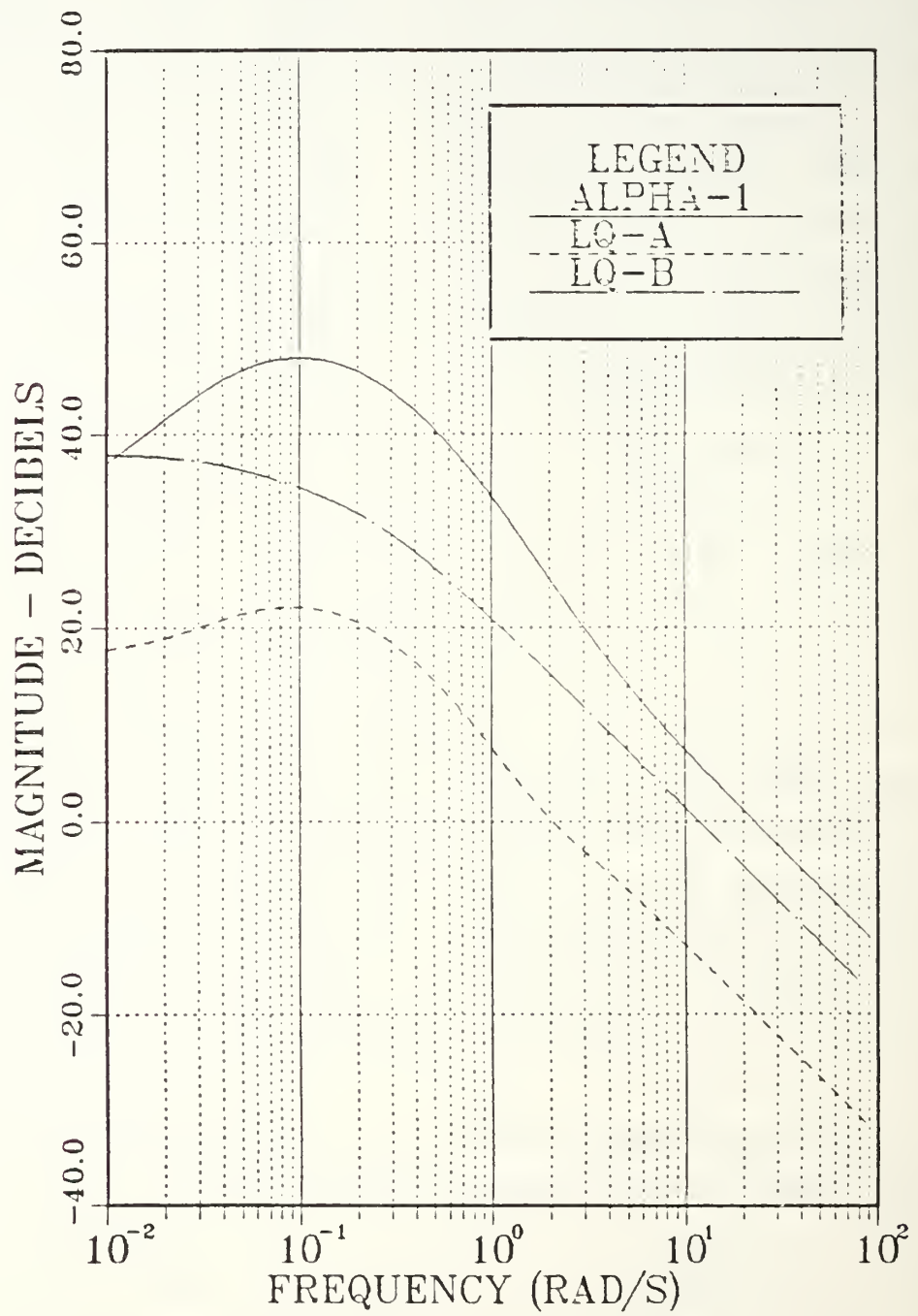


Figure 5.20 Bode Plots Comparison- Input 1-1.

OPEN LOOP GAIN 1-2

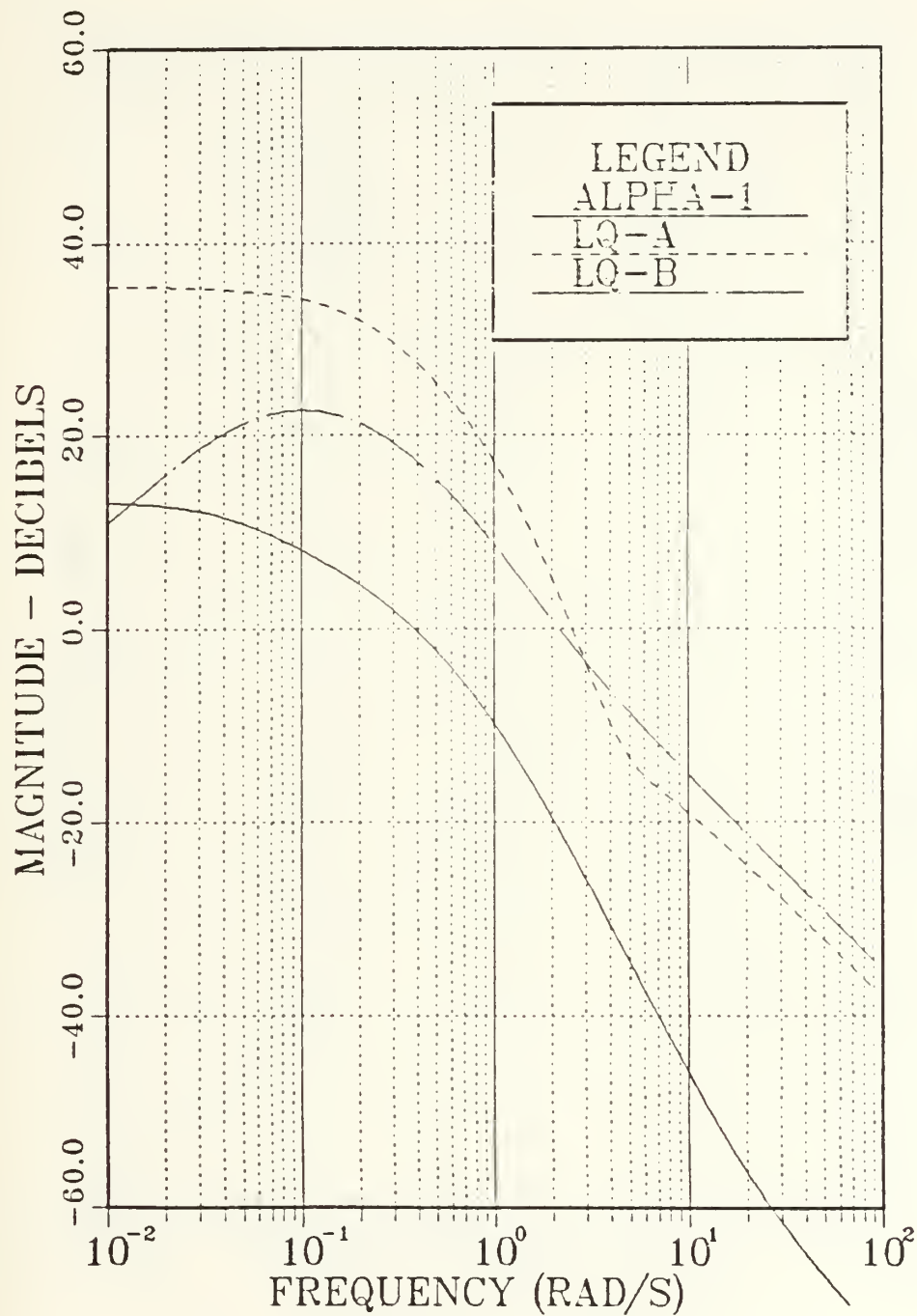


Figure 5.21 Bode Plots Comparison - Input 1-2.

OPEN LOOP GAIN 2-1

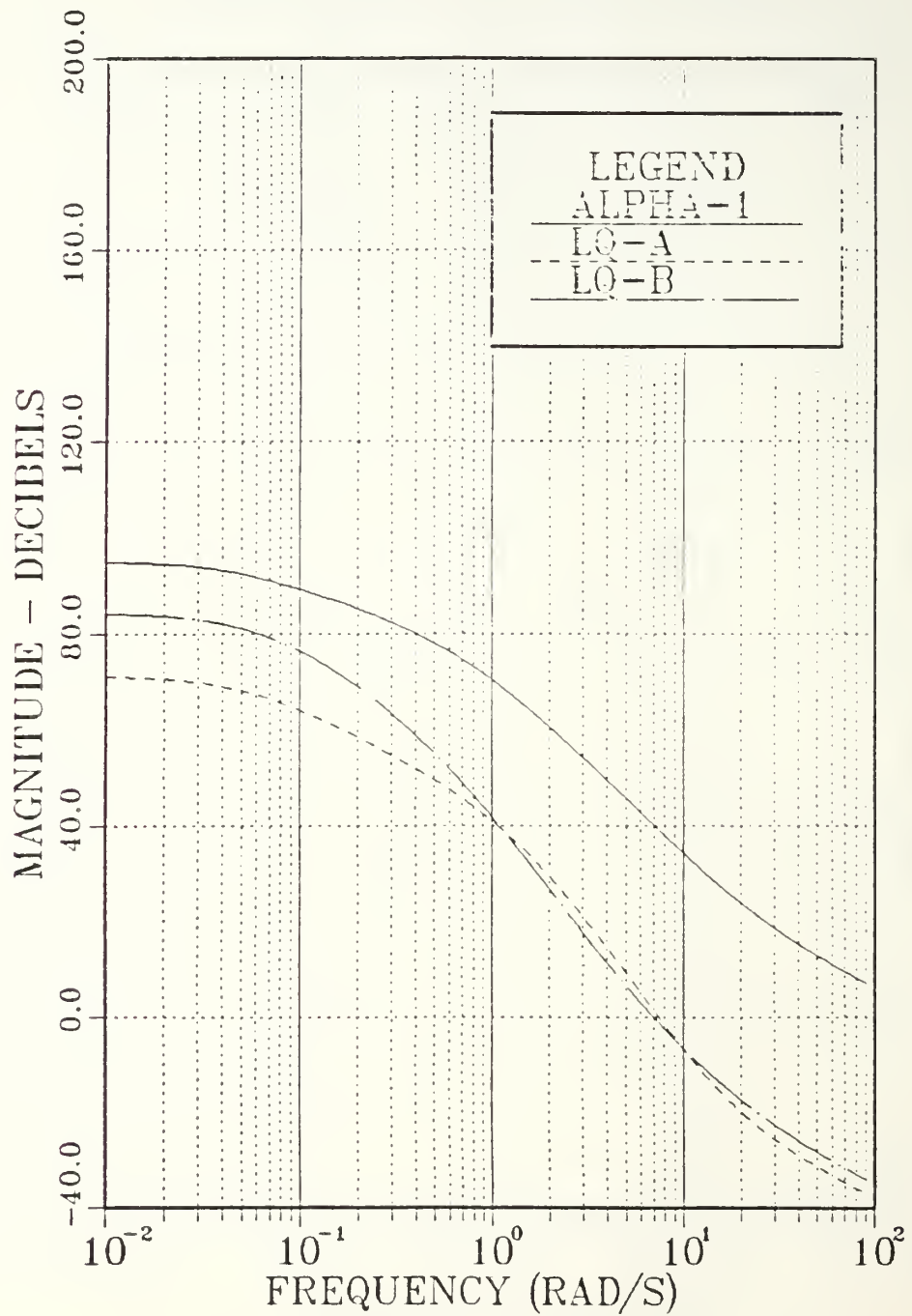


Figure 5.22 Bode Plots Comparison - Input 2-1.

OPEN LOOP GAIN 2-2

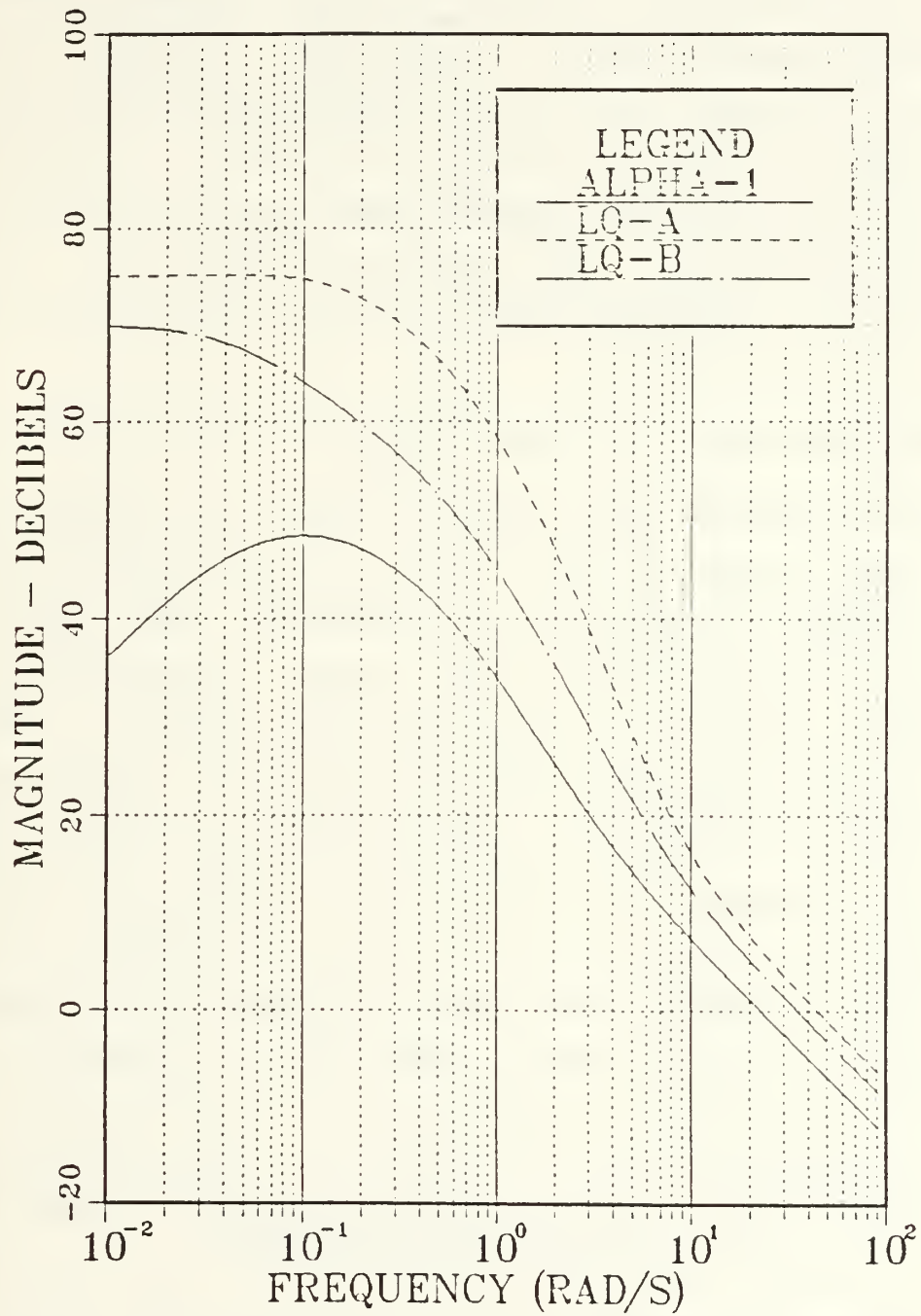


Figure 5.23 Bode Plots Comparison - Input 2-2.

to be an acceptable design but was shown to have undesired transmission zeros in other channels; this can also be seen from the closed-loop plot for the $r-\phi_c$ channel in Figure 5.31. Design LQ-A, obtained from the reassignment sequence A, is characterized by very low DC gain (-9dB) and 'peak' in frequencies near 10 rad/s. These have been shown, both in the time response and pole-zero discussions that it will have undesired effect on the pilot's control. Design LQ-B, obtained from the second reassignment sequence appears to be the best compromise. All channels (Figures 5.30 to 5.32) have flat low frequency characteristic and coupling between modes are almost absent.

C. DISCUSSION AND CONCLUDING REMARKS

A new computer aided design procedure for the multivariable linear time-invariant system using Linear Quadratic Pole Placement formulation is presented. The two design examples presented above have served to demonstrate the complexity involved in a multivariable design. The main problem lies in the fact that the solution of a MIMO problem is in general non-unique. This was shown in the helicopter problem where different approaches result in different designs, although the closed-loop eigenvalues for all designs were the same. It was also shown that the extra degrees of freedom in MIMO system design can be accounted for by analyzing the singular value plots, transmission zero movements and closed-loop eigenvectors of the designs. It becomes apparent that the success of any multivariable design methodology hinges on how to make use of these extra degrees of freedom available in MIMO system. It was also shown in the above design examples that the Linear Quadratic Pole Placement procedures possess such quality. Some unique properties of the method presented here are as follows;

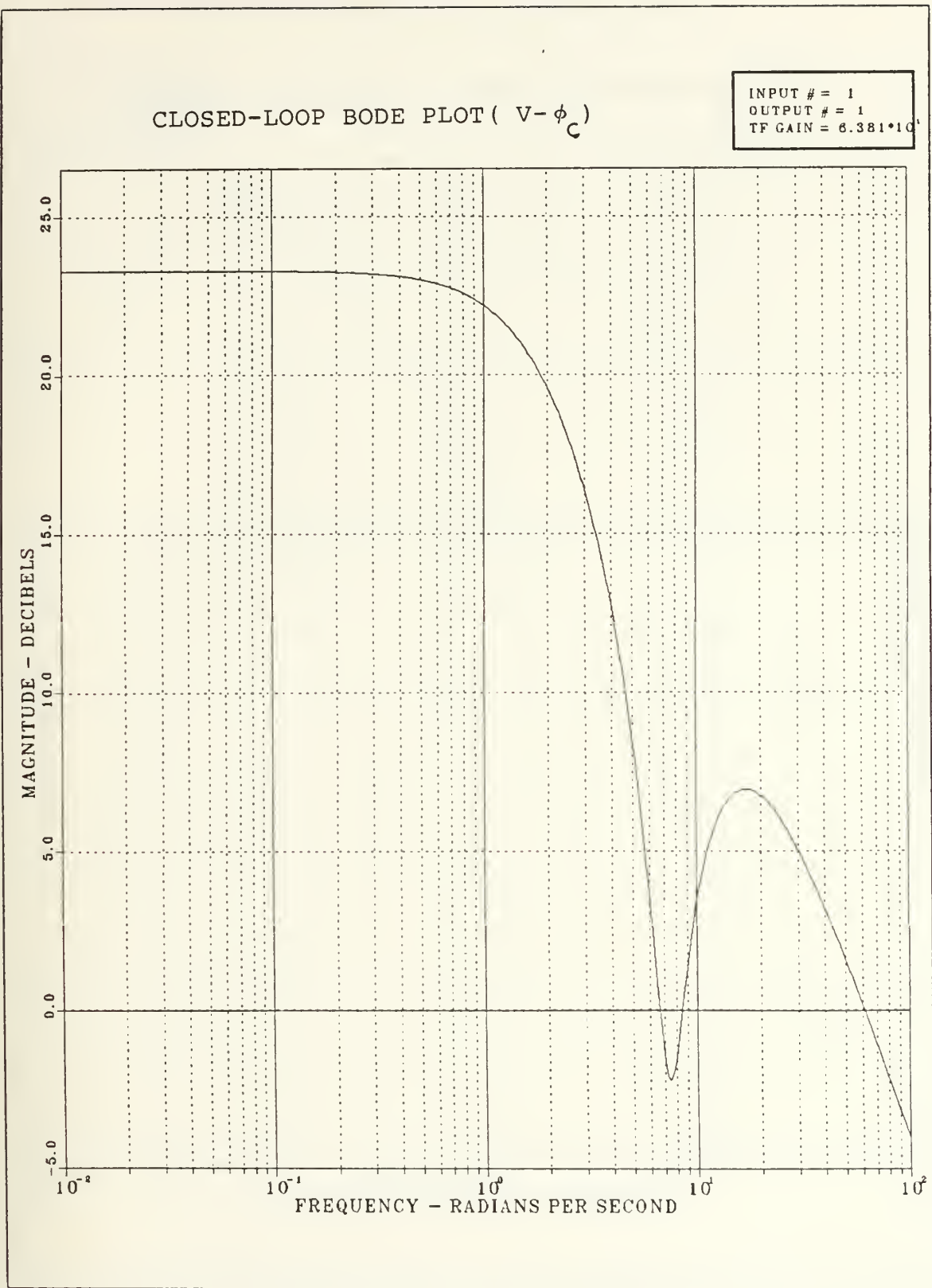


Figure 5.24 Closed-Loop Bode Plot- AlphaTech 1-1.

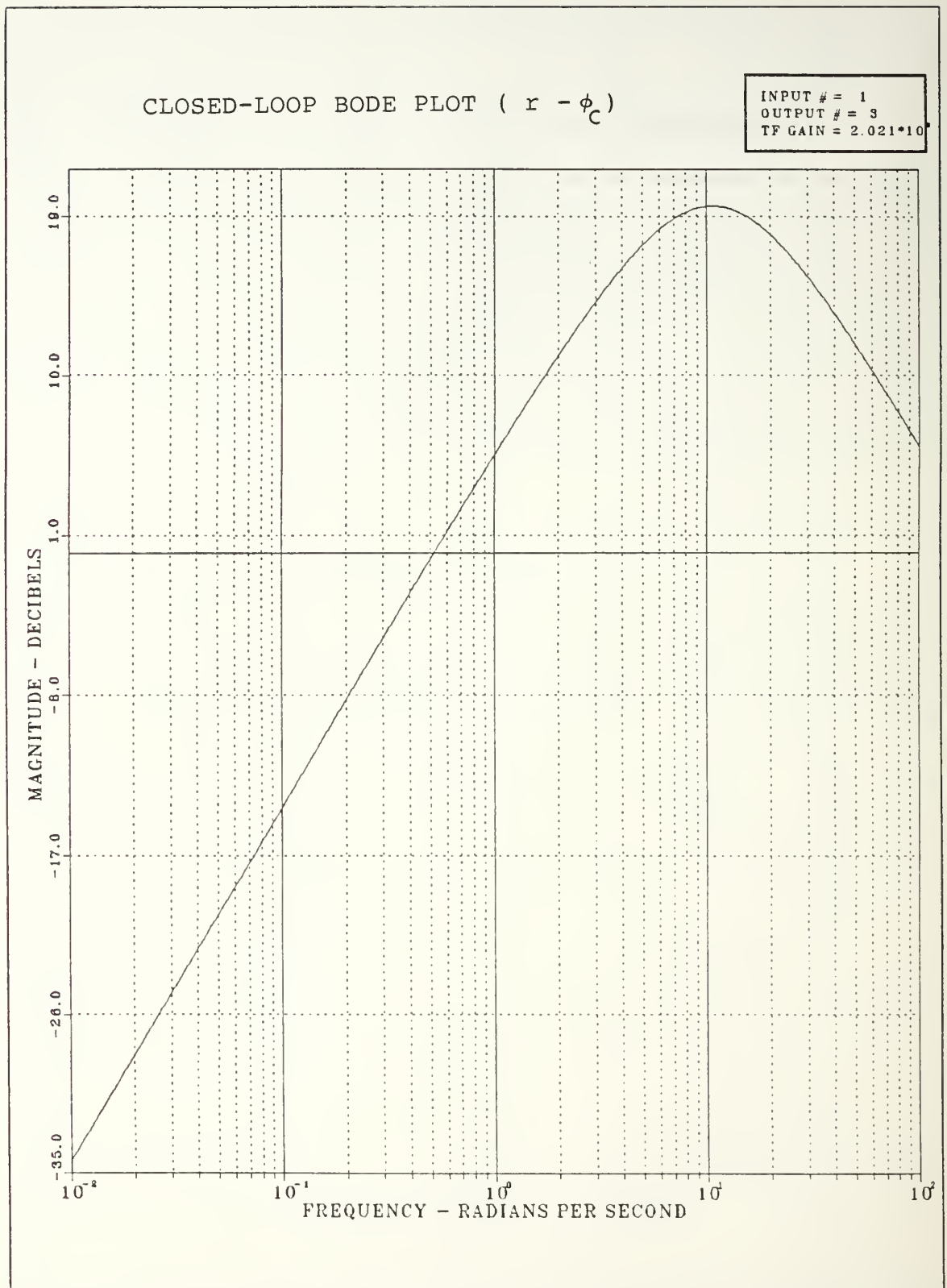


Figure 5.25 Closed-Loop Bode Plot- AlphaTech 1-3.

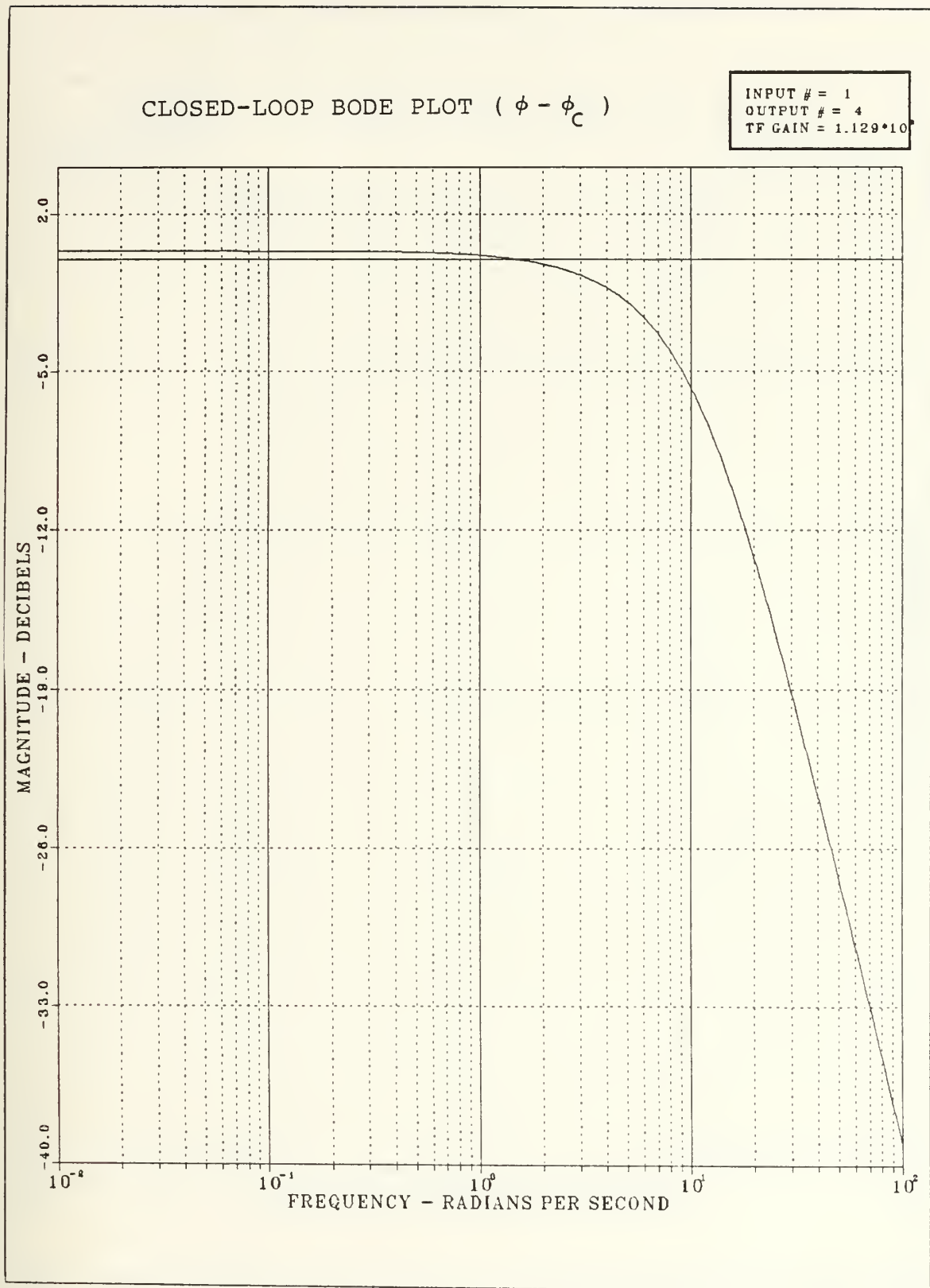


Figure 5.26 Closed-Loop Bode Plot- AlphaTech 1-4.

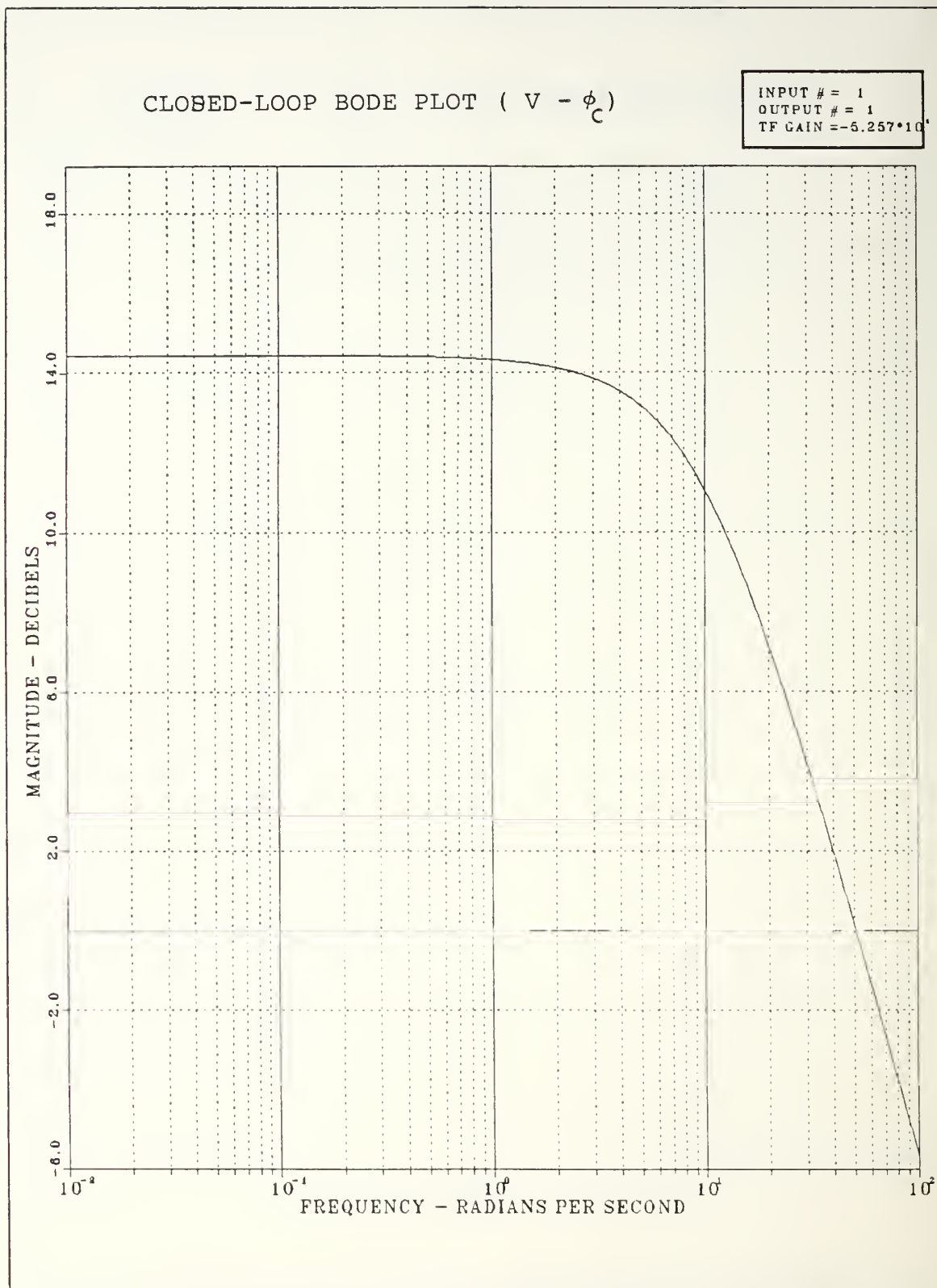


Figure 5.27 Closed-Loop Bode Plot- LQ-A 1-1.

CLOSED-LOOP BODE PLOT ($r - \phi_c$)

INPUT # = 1
 OUTPUT # = 3
 TF GAIN = -1.386*10⁻¹

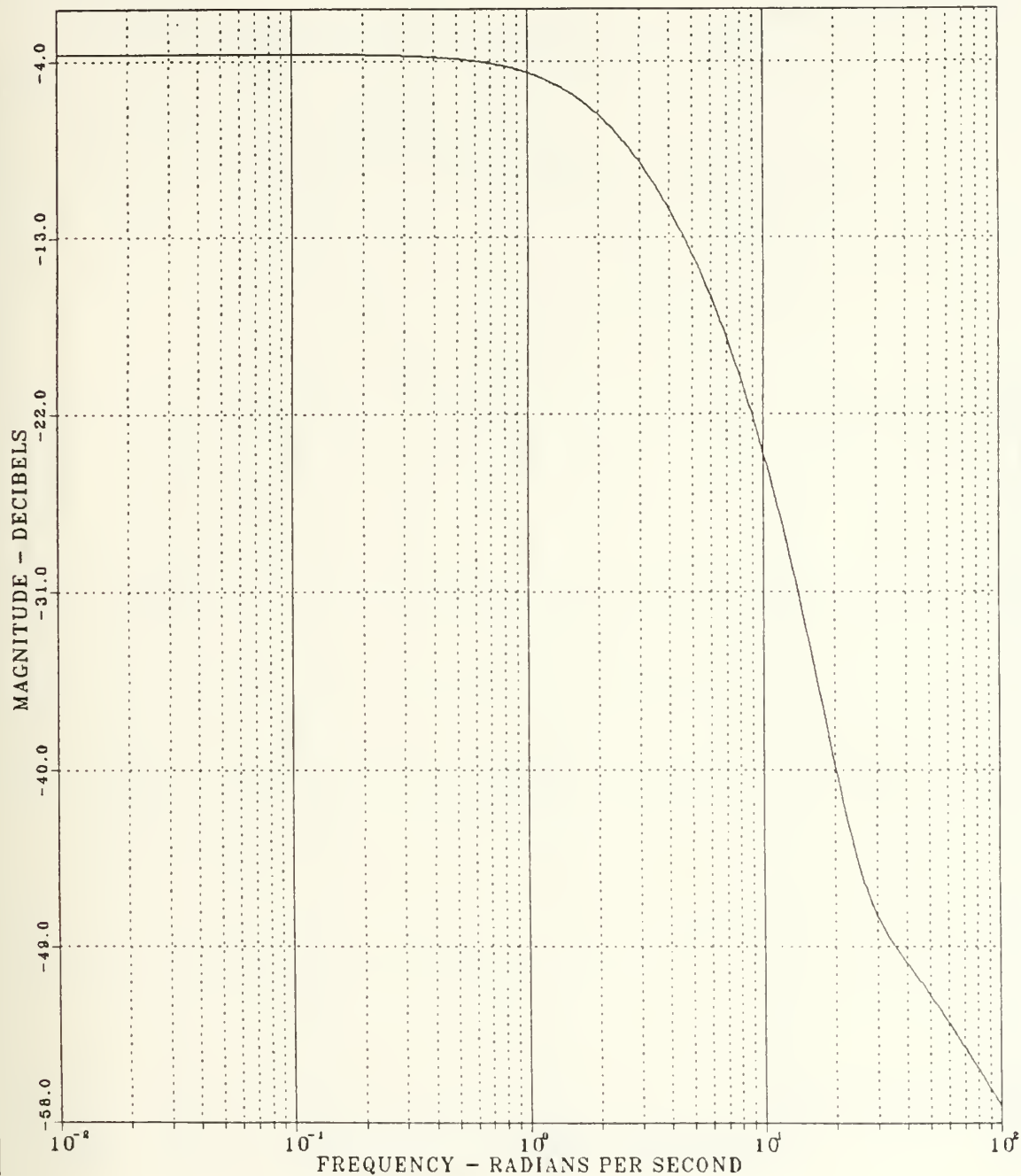


Figure 5.28 Closed-Loop Bode Plot- LQ-A 1-3.

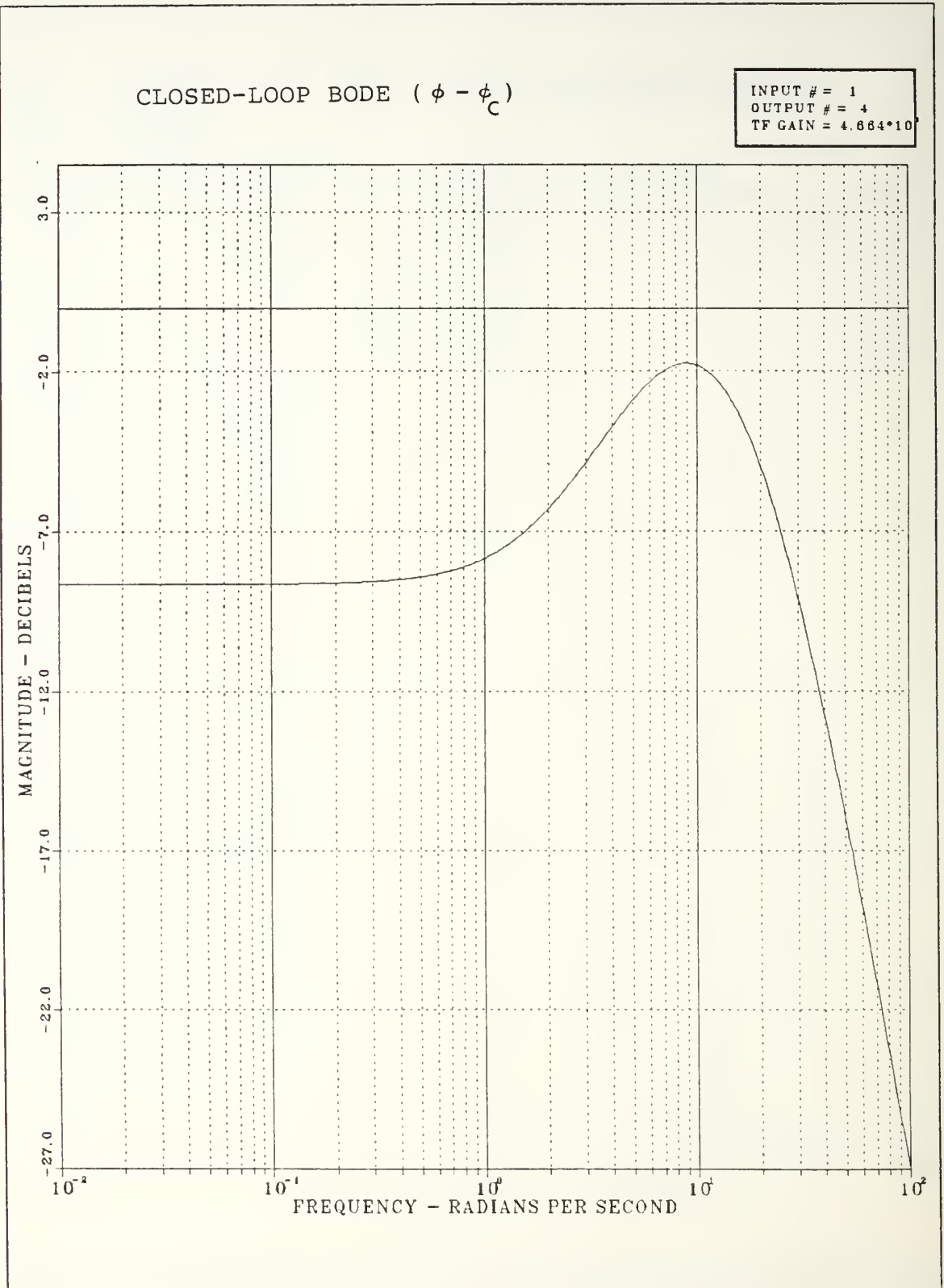


Figure 5.29 Closed-Loop Bode Plot- LQ-A 1-4.

CLOSED-LOOP BODE PLOT ($v - \phi_c$)

INPUT # = 1
 OUTPUT # = 1
 TF GAIN = $-2.831 \cdot 10^1$

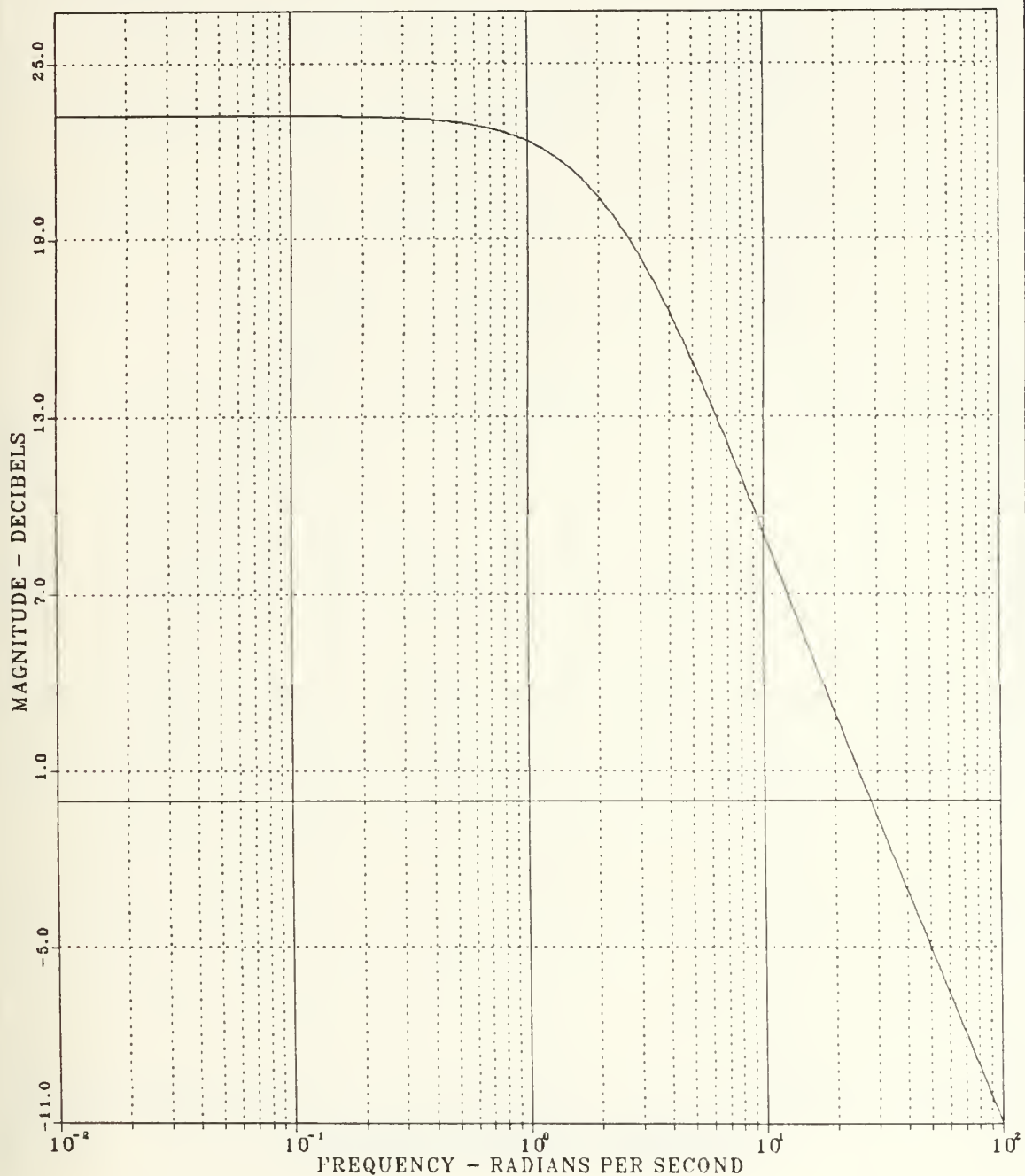


Figure 5.30 Closed-Loop Bode Plot- LQ-B 1-1.

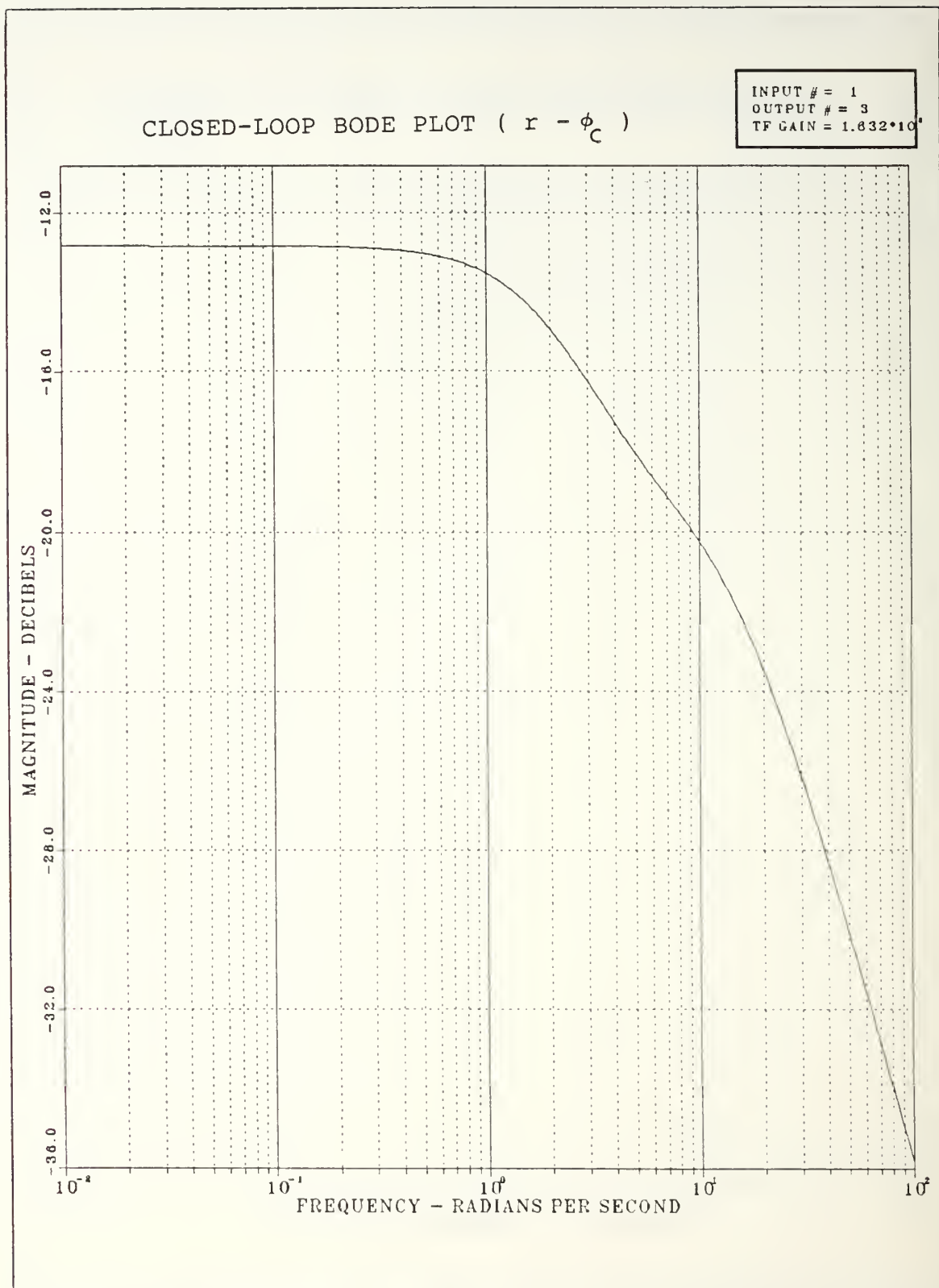


Figure 5.31 Closed-Loop Bode Plot- LQ-B 1-3.

CLOSED-LOOP BODE PLOT ($\phi - \phi_C$)

INPUT # = 1
OUTPUT # = 4
TF GAIN = $2.560 \cdot 10^7$

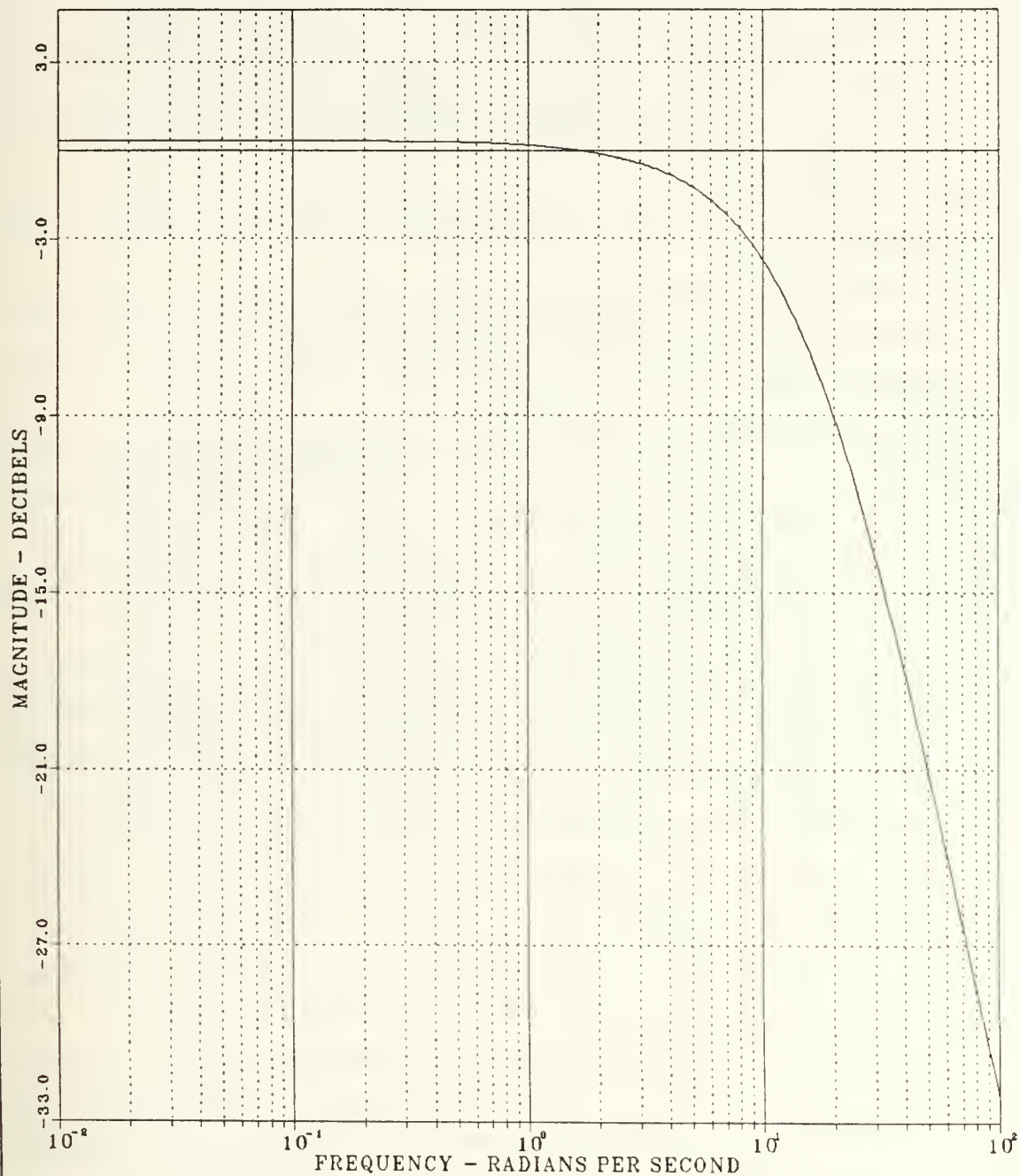


Figure 5.32 Closed-Loop Bode Plot- LQ-B 1-4.

1. It guarantees stability automatically.
2. It has partly overcome the main difficulty in applying optimal control theory to practical problems; i.e. that of selecting suitable performance indices (Q and R matrices). The designer can now choose to compute Q (if he knows something about R) or vice versa or compute both Q and R if both of them are unknown.
3. It has built-in robustness to model's error and perturbation.
4. Reduced-order type of problem formulation is possible as the procedure can reassign poles one at a time without affecting others.

On the other hand, the design procedure developed here is not without some shortcomings. Three main problem areas are described below, one of which can be overcome by additional programming efforts while the others are still active research areas pursued by many researchers.

1. Eigenvectors Assignments: It was shown in the design examples presented above that while the overall speed of response of the closed-loop system is determined by its eigenvalues, the 'shape' of the transient response depends on the closed-loop eigenvectors. The problem of eigenvectors assignments was first formulated in [Ref. 16]. Since then many eigenvectors assignment algorithms have been developed [Refs. 17,24] for use in multivariable design. In principle, the eigenvectors assignment routine can be incorporated into the coordinate transformation portion of the OPTPP program (as indicated in Figure 4.1 in Chapter 4). In essence, a new similarity transformation different from M in Equation 4.6 is computed once the eigenvalues and eigenvectors are specified. As most eigenvector algorithms available

at present are iterative in nature and their inclusion requires major programming efforts, it is recommended for future work.

2. Perturbation or Model's Error Structure: The guaranteed robustness obtained from the LQ formulation given by equations 3.4 and 3.5 ensures that the perturbation or model's error ($L(j\omega)$) is sufficiently small so that the closed-loop system remains stable. However, the above only applies to simple model's error structure where both $L(j\omega)$ and R are diagonal matrices. There may be cases when the above equations do not hold and hence the design becomes very conservative. An example of this nature is shown in [Ref. 19]. An ad hoc solution is to use non-diagonal control weighting matrix as mentioned in Chapter 4. Two designs for the helicopter problem are obtained using non-diagonal R . Their results in terms of singular value plots, Bode plots etc are compared with other designs in Appendix A
3. Reassignment Sequence: As shown in the last section, different reassignment sequences result in different designs. At present there are no known methods for determining the best reassignment sequences.

VI. CONCLUSIONS

It was demonstrated that the general pole assignment problem in multivariable state feedback control system design can be formulated using the Linear Quadratic Control approach. This method of formulation is effective for two main reasons; First, the extra degrees of freedom available in a multivariable system structure is utilized to produce designs that are robust to perturbations in the system and gain matrices. Secondly, the classical difficulty of selecting suitable performance index in optimal control formulation was partly overcome, as designers now have the flexibility of specifying only Q (the state weighting matrix) or R (the control weighting matrix) or both Q and R as the design parameters to be varied. In other words, knowledge of the performance index which ideally should come from physical argument is used to the best of designer's advantage. In addition, the structure of the present formulation is such that eigenvector assignment can be further incorporated into the procedure. The above properties, when combined with the reduced-order formulation capability, have been shown to be very versatile and have important impact on the performance of the resulting design.

The optimal pole placement (OPTPP) program developed here is combined with other well established routines to form a computer aided design and synthesis package. Together with the design procedure and philosophy presented here, it provides the control system designer an excellent and viable tool to solve complex multivariable problems.

The procedure was applied to practical test examples, and numerical results were presented and discussed. Results indicated that all controllers obtained from the formulation

given here were stable and robust. Introducing perturbation in the system matrices leaded only to small errors in the assigned poles. The main shortcoming of the design procedure is the ad hoc nature in which the poles are reassigned. More research is required to develop a systematic way of assigning poles and its eigenvectors thus allowing the designer to optimally shape the response of the system.

APPENDIX A

NON-DIAGONAL R DESIGNS

The two design examples presented in Chapter 5 were based on diagonal control weighting matrix (i.e. $R = I$). It is now illustrated that design using non-diagonal R will provide yet another degrees of freedom available to the designer. This type of formulation is especially useful when the model's or error structure is known. For multiplicative type of perturbation, the effect of control weighting matrix on the system stability has been explored in [Ref. 12]. In essence, the selection of the R matrix determines the coordinate frame in which the sensitivity assessment is to be made. This can be readily seen from equation 3.1, the general form in which R is non-diagonal is given below;

$$\bar{\sigma} [R^{1/2} L^{-1}(s) R^{-1/2} - I] < \underline{\sigma} [I + G(s)] \quad (\text{eqn A.1})$$

In general, the sensitivity of the system to perturbation in a particular $L(s)$'s direction can be reduced by making $\bar{\sigma} [R^{1/2} L^{-1}(s) R^{-1/2} - I]$ small. This can be done simply by choosing an appropriate R . The main problem with this kind of approach is that $L(s)$ must be known precisely for all frequency for the computation of $\bar{\sigma} [R^{1/2} L^{-1}(s) R^{-1/2} - I]$. In addition, the worst case direction of $L(s)$ must be known otherwise the resulting design may be too conservative.

The helicopter problem presented in Chapter 5 is now analyzed using the non-diagonal control weighting matrix. Two designs are obtained as follows;

Design one (LQ-C):

$$R = \begin{bmatrix} 1.0 & 0.0 \\ 0.0 & 2.0 \end{bmatrix}$$

Design two (LQ-D):

$$R = \begin{bmatrix} 1.0 & 0.5 \\ 0.5 & 1.0 \end{bmatrix}$$

Design LQ-C is obtained using an assignment sequence similar to LQ-A with no off-diagonal element in R. Design LQ-D is obtained using placement sequence for design LQ-B but has off-diagonal element in R. Q and F obtained during each move and the final Q_e and F_e are tabulated in Tables VII and VIII. Singular value plots and the open-loop Bode Plots are compared with other designs in Figures A.1 to A.5

Both design has singular value greater than one for all frequency. It is interesting to note that similar reassignment sequence produce similar singular value plots. This can be readily seen by comparing LQ-A with LQ-C and LQ-B with LQ-D in the figures. The effect of using non-diagonal R merely changes the shape of the singular value plots. For the cases presented here, both non-diagonal R designs are slightly less conservative (having lower singular value).

As in the singular value plots, the shape of the open-loop Bode plots (with feedback) are closely related to the reassignment sequence. (compare Bode plots of LQ-A with LQ-C and LQ-B with LQ-D in Figures A.2 to A.5 . Results from pole-zero maps also indicate similar trends.

In summary, it is demonstrated that designs using non-diagonal control weighting matrix (including possibly off-diagonal elements) provide yet another means of 'fine

tuning' the design. This capability, such as using the off-diagonal elements directly in design, is unique to the present formulation. Robustness between the upper and lower crossfeed (of $L(s)$) can be controlled by adjusting the relative weighting of the upper and lower (or off-diagonal) elements of the control weighting matrix. It must be emphasized that this kind of fine tuning is only possible for a class of rather well-defined structure of L . In practices, other constraints, such as the energy of the control input, the conditioning of R etc must be considered over the range of frequencies. The key issue of model's error structure and how it can be used in multivariable control system design is currently being investigated by many researchers.

TABLE VII
RESULTS FROM POLE PLACEMENT SEQUENCE (LQ-C)

Move	Q and F obtained during each reassignment			
Q ₁	0.00062	0.10604	0.01180	0.09584
	0.10604	18.17288	2.02209	16.42484
	0.01180	2.02209	0.22500	1.82758
	0.09584	16.42485	1.82758	14.84496
F ₁	0.00033	0.05708	0.00635	0.05159
	-0.01773	-3.03892	-0.33814	-2.74657
Q ₂	4.17258	-3.78394	0.13594	-108.79945
	-3.78394	3.43150	-0.12328	98.66574
	0.13594	-0.12328	0.00443	-3.54455
	-108.79944	98.66574	-3.54455	2836.93335
F ₂	0.02701	-0.02449	0.00090	-0.70429
	1.21663	-1.10329	0.03967	-31.72316
Q ₃	154.23135	36.25874	26.52098	418.56201
	36.25874	8.52419	6.23491	98.40111
	26.52100	6.23491	4.56044	71.97421
	418.56201	98.40114	71.97421	1135.91846
F ₃	2.80842	0.66025	0.48291	7.62180
	-7.58133	-1.78234	-1.30355	-20.57501
Q ₄	0.44288	0.09189	-4.22665	0.82784
	0.09189	0.01906	-0.87692	0.17176
	-4.22665	-0.87692	40.33727	-7.90052
	0.82784	0.17176	-7.90052	1.54741
F ₄	-0.63449	-0.13165	6.05549	-1.18605
	-0.09950	-0.02064	0.94961	-0.18599
F _e	2.20127	0.56119	6.54565	5.78305
	-6.48193	-5.94519	-0.65241	-55.23073
Q _e	158.84741	32.67273	22.44206	310.68604
	32.67273	30.14761	7.25680	213.66344
	22.44208	7.25680	45.12712	62.35670
	310.68604	213.66348	62.35670	3989.24390

$$u(t) = -F x(t) + h \phi_c(t), \quad h = \begin{bmatrix} 5.7839 \\ -55.23 \end{bmatrix}$$

TABLE VIII
RESULTS FROM POLE PLACEMENT SEQUENCE (LQ-D)

Move	Q and F obtained during each reassignment			
Q_1	0.00023	0.03940	0.00438	0.03561
	0.03940	6.75195	0.75129	6.10249
	0.00438	0.75129	0.08360	0.67902
	0.03561	6.10249	0.67902	5.51550
F_1	0.00895	1.53374	0.17065	1.38619
	-0.01765	-3.02506	-0.33659	-2.73405
Q_2	0.24621	0.11866	14.98355	9.73271
	0.11866	0.05719	7.22105	4.69050
	14.98355	7.22105	911.83813	592.29370
	9.73272	4.69050	592.29370	384.73022
F_2	0.56910	0.27424	34.63293	22.49620
	-0.32115	-0.15476	-19.54404	-12.69504
Q_3	0.17288	-0.43406	7.14036	-7.87012
	-0.43406	1.08987	-17.92834	19.76067
	7.14036	-17.92834	294.92139	-325.06299
	-7.87012	19.76067	-325.06299	358.28516
F_3	0.05761	-0.14465	2.37951	-2.62273
	0.34047	-0.85487	14.06264	-15.49990
F_e	0.63566	1.66333	37.18307	21.25964
	0.00167	-4.03469	-5.81798	-30.92899
Q_e	0.41932	-0.27600	22.12828	1.89820
	-0.27600	7.89901	-9.95601	30.55365
	22.12828	-9.95601	1206.84302	267.90967
	1.89821	30.55365	267.90967	748.53076

$$u(t) = -F_x(t) + h \phi_c(t), \quad h = \begin{bmatrix} 21.159 \\ -30.9289 \end{bmatrix}$$

SINGULAR VALUE PLOT

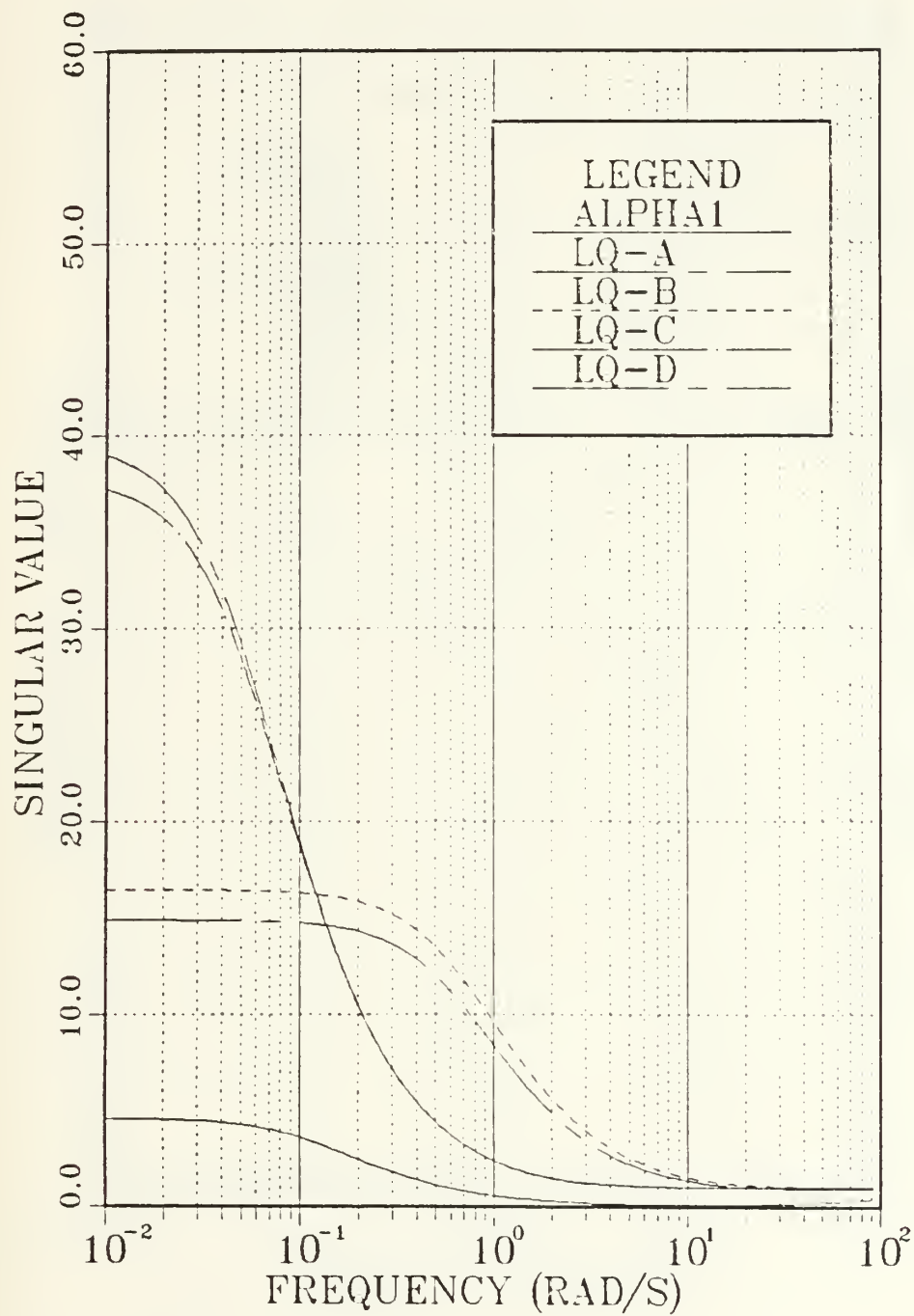


Figure A.1 Singular Value Plots - Comparison.

OPEN LOOP GAIN 1 - 1

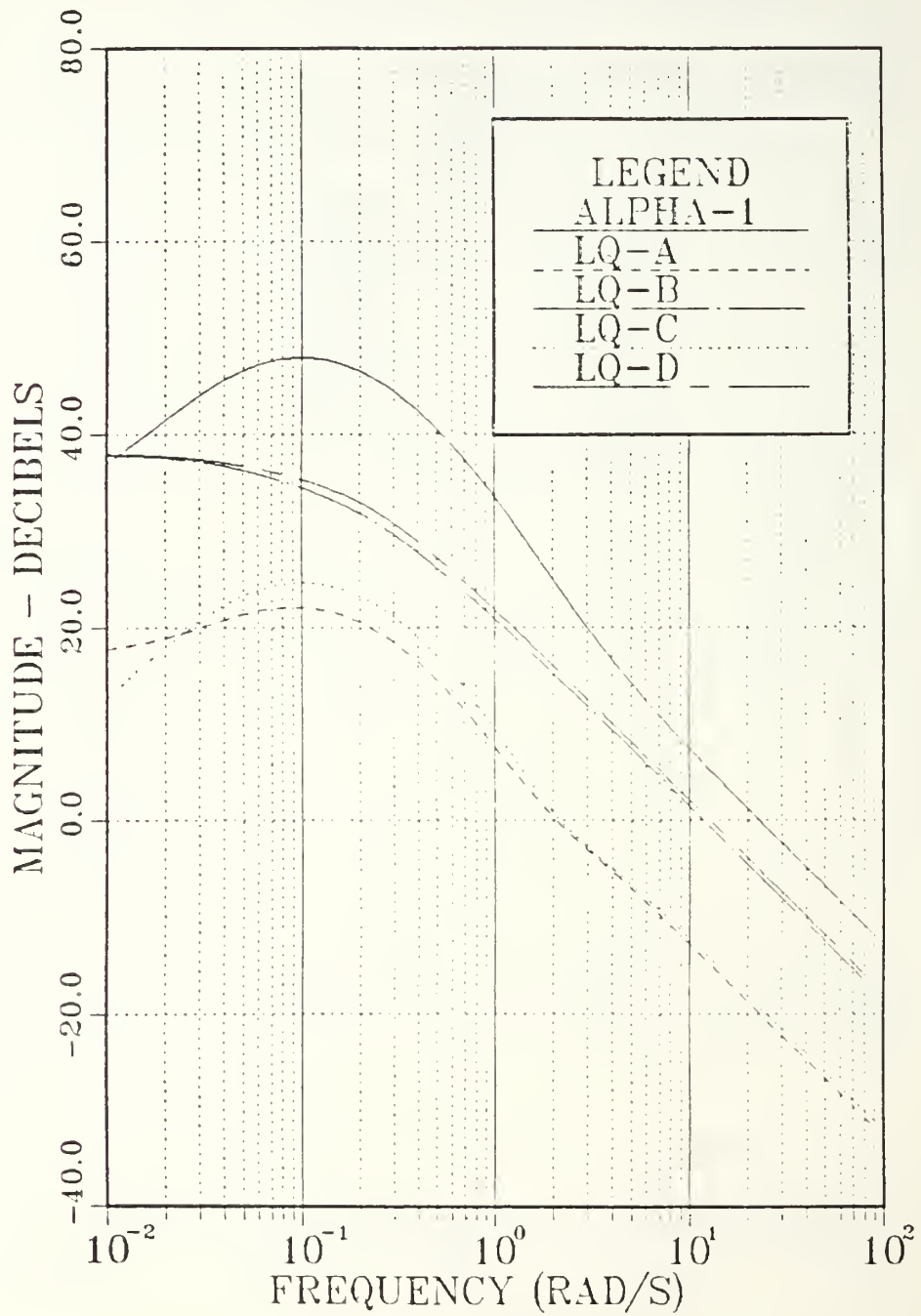


Figure A.2 Bode Plots Comparison- Input 1-1.

OPEN LOOP GAIN 1-2

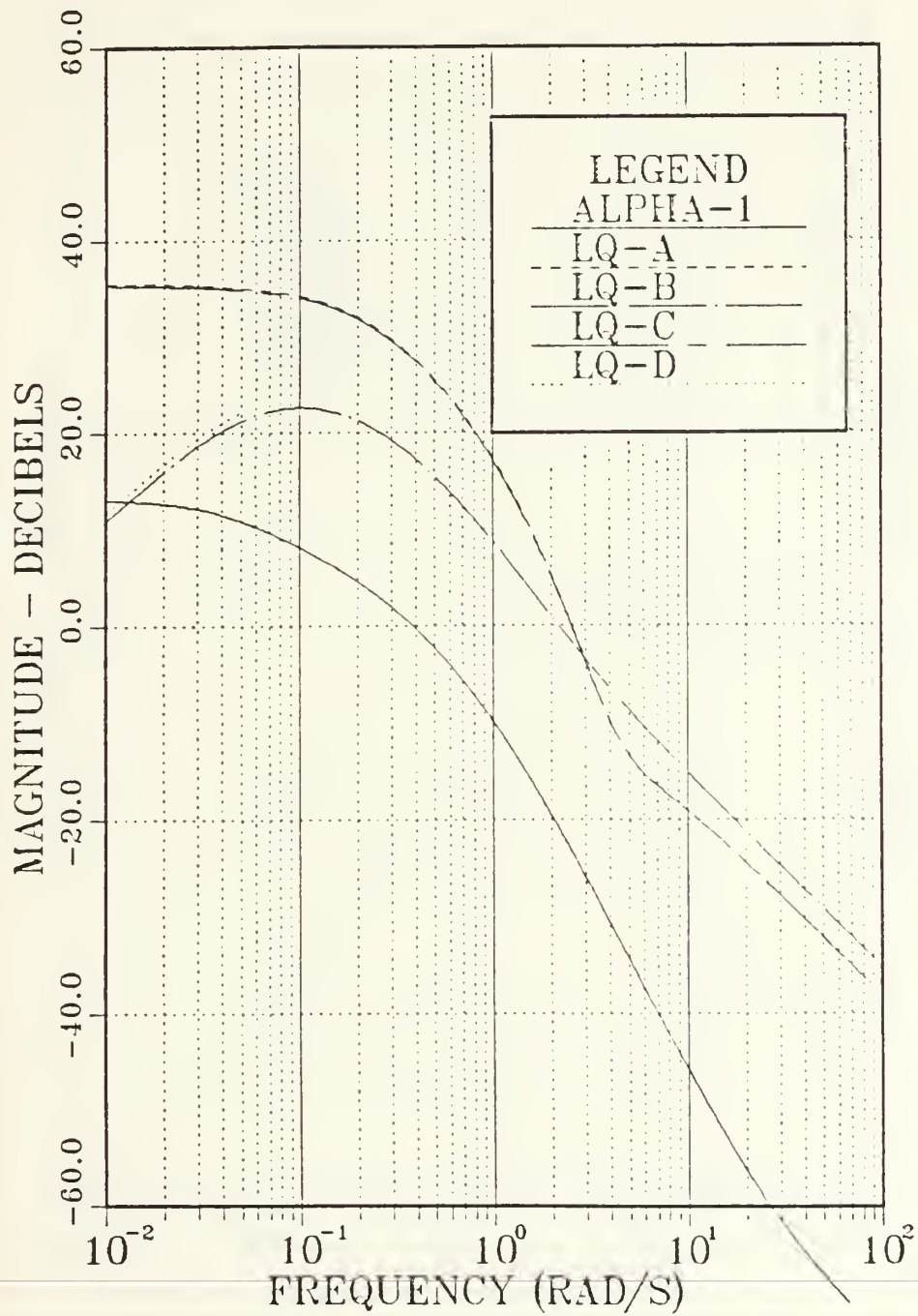


Figure A.3 Bode Plots Comparison - Input 1-2.

OPEN LOOP GAIN 2-1

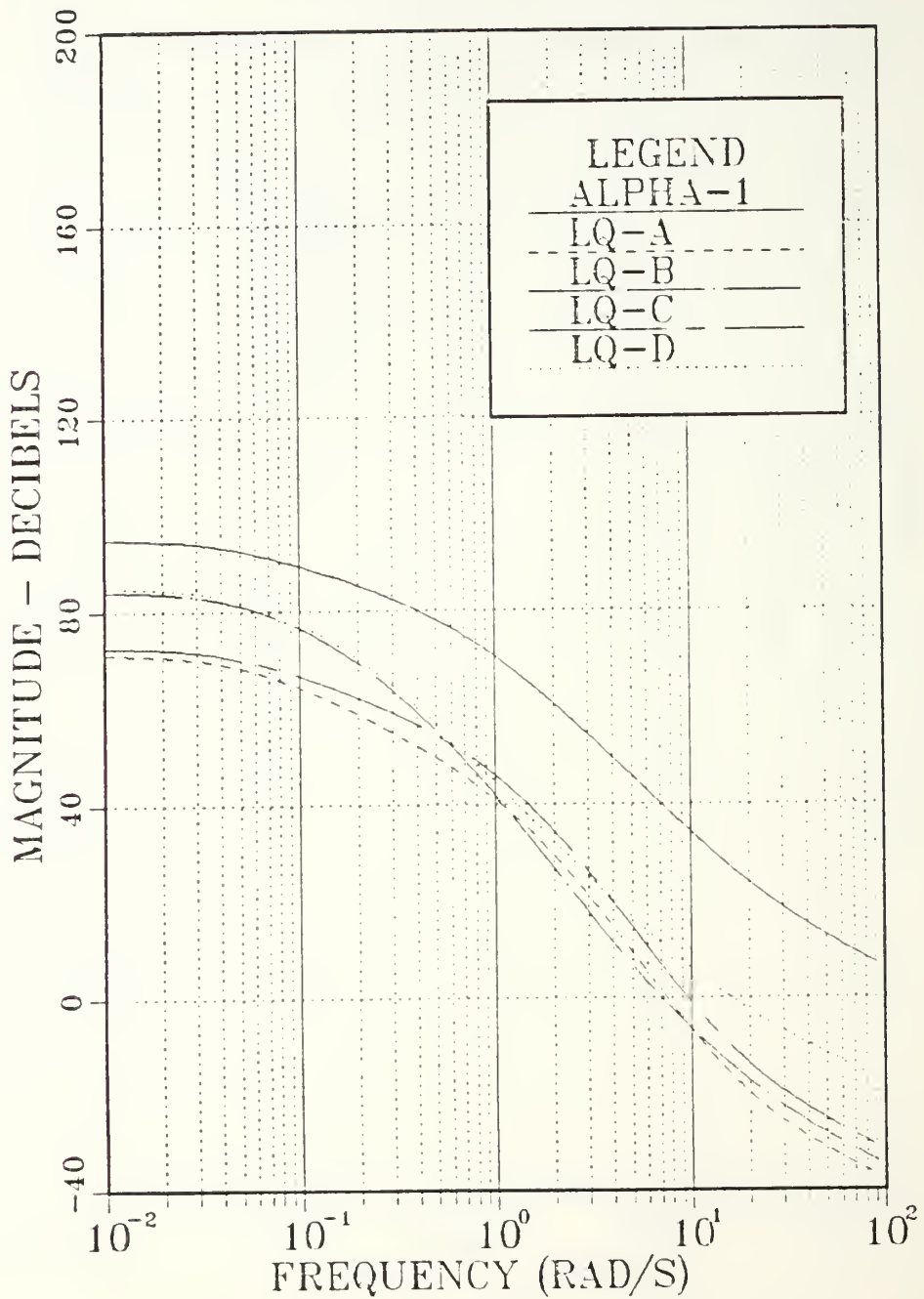


Figure A.4 Bode Plots Comparison - Input 2-1.

OPEN LOOP GAIN 2-2

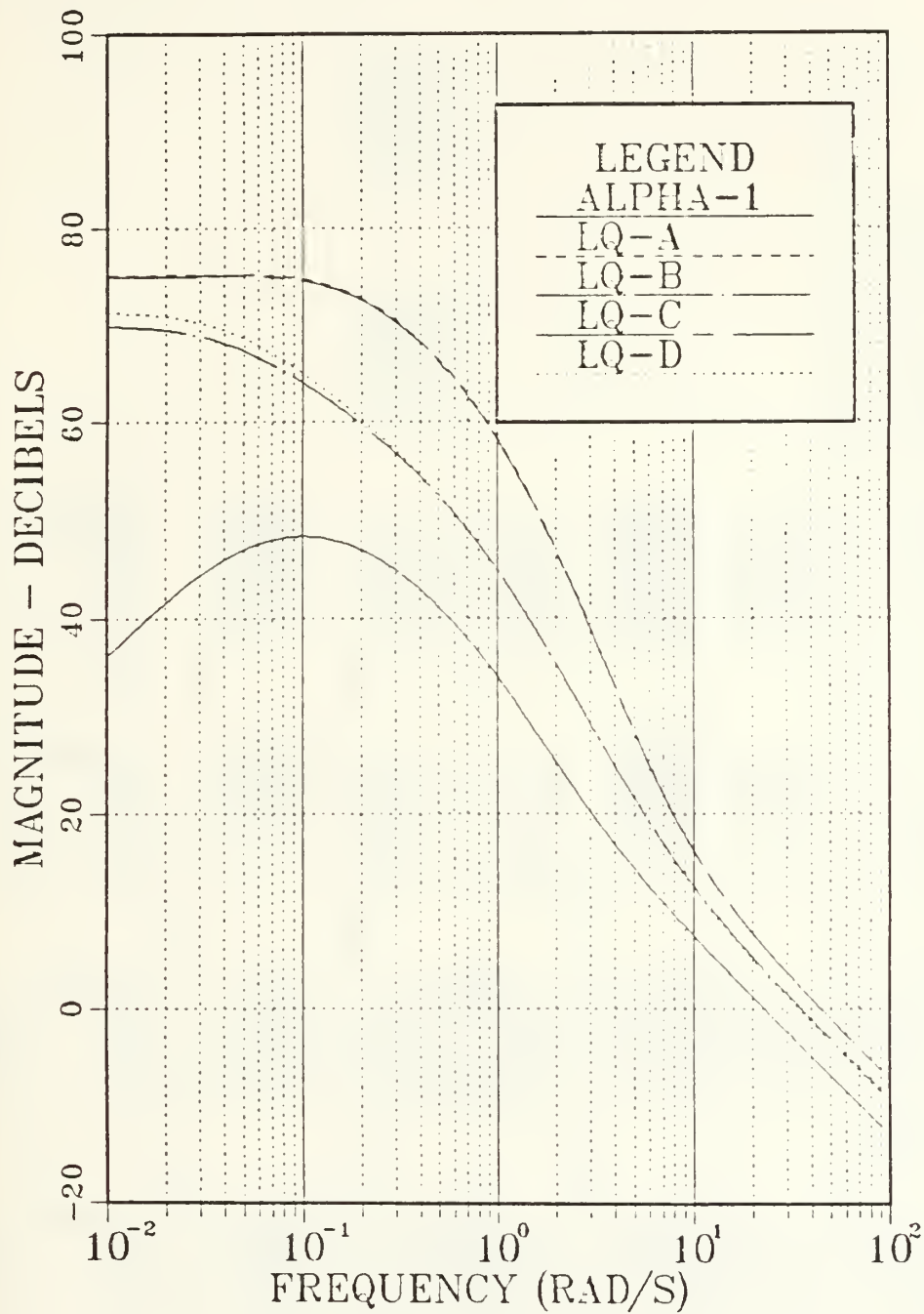


Figure A.5 Bode Plots Comparison - Input 2-2.

APPENDIX B : EXAMPLE DESIGN RUN OUTPUTS

```

** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** 
THIS EXAMPLE DEMONSTRATES THE POLE REASSIGNMENT WHERE THE OPEN
LOOP POLE AT -2.126 IS MOVED TO ITS NEW LOCATION AT -25.21
6(4, 4) IS THE DESIGN VARIABLE, NOTE THAT THE UNSTABLE POLES
0.2005 IS AUTOMATICALLY MOVED TO ITS MIRROR IMAGE DUE TO LQ
FORMULATION.
** ** ** ** 

```

```

CONTROL          0
*** THE A PLANT MATRIX ***
-2.27000      1.42000      -0.15000      31.99001
 0.04000      -0.70000      -0.07000      0.00000
 0.04000      -0.05000      -0.05000      0.00000
 0.00000      1.00000      0.11000      0.00000

```

```
*** THE B CONTROL INPUT MATRIX ***
```

6.12000	6.95000
0.04000	-8.37000
0.34000	0.02000
0.00000	0.00000

THE STARTING Q WEIGHTING MATRIX ***[illegible]

```

**
**
**
THE DESIGN VARIABLE(C) X
**
**
**

```

0000

0000

0000

0000

*** THE R WEIGHTING MATRIX ***

1.00000	0.00000
0.00000	1.00000

*** THE STARTING/FINAL F MATRIX ***

-14.78000	2.16000	77.97000	-55.30000
-0.00567	-2.55600	0.39400	-15.05000

*** THE ORDERED COMPLEX EIGENVALUES (INPUT)

-25.21001 C.00000

** ICOMP ** 0

AUX TRANSFORMATION L

*
**

DIRECTOR OF A
GENEVE

*
**

0.14268	-0.98509	0.99767
0.10820	-0.00101	-0.03687
-0.92384	0.15380	-0.04176
0.00049	0.07703	0.03949

*** TRANSFORMED A MATRIX

-	0.05030	0.00000	0.00000
0.00000	0.20653	0.00000	0.00000
0.00000	-	0.00001	-1.04987
0.00008	0.00001	0.00001	0.00002

TRANSCURED
C. MATRIX

[illegible]

TRANSEUROPEAN
C. MATRIX

-0.15647
 -0.85855
 -1.80459
 -1.09721

*** I.09121 RCOT INVERSE ***
*** SQUARE ***
*** -95.12333 ***
*** CF R ***

SPEAK	KCU	INVEST
1.00000	0.00000	0.00000
0.00000	1.00000	1.00000

0.999777
-0.007965
-0.019455
0.00475

0.0000
0.0000
0.0001
-2.12635

○○○○
●●●●
▽▽▽▽

```

AAAAA DDDDD SSSSS
A  A  D  D  S
A  A  D  D  S
AAAAAA D  D  SSSSS
A  A  D  D  S
A  A  D  D  S
A  A  D  D  SSSSS
A  A  D  D  SSSSS

```

F O R T R A N P R O G R A M
 F O R
 A U T O M A T E C D E S I G N S Y N T H E S I S
 V E R S I O N 1.00

```

CONTROL PARAMETERS      = 3      IONED = 3      IPRINT = 1000
ISTRAT = 5              = 1      NCCN  = 2
IGRAD  = 0              NDV

```

1

 OPTIMIZATION RESULTS

OBJECTIVE FUNCTION VALUE 0.56055E-13

DESIGN VARIABLES

	LOWER BOUND	VALUE	UPPER BOUND
VARIABLE 1	0.00000E+00	0.68858E-01	0.10000E+05

DESIGN CCNSTRANTS

1) -C.50C0E-01 -C.9000E+00

FUNCTION EVALUATIONS = 41

*** THE OPTIMAL Q MATRIX

0.00000	0.00000	0.00000	0.00000
0.00000	0.00000	0.00000	0.00000
0.00000	0.00000	0.00000	0.00000
0.00000	0.00000	0.00000	0.06886

END OF POLE PLACEMENT ROUTINE

THE Q REQUIRED FOR THIS ASSIGNMENT IS

0.10027	0.96194	0.11959	-1.50849
0.96194	9.22847	1.14725	-14.47182
0.11959	1.14725	0.14262	-1.79908
-1.50849	-14.47182	-1.79908	22.69429

EIGENSYSTEM OF OPTIMAL REGULATOR.....

C-LOOP OPTIMAL REG. E-VALUES...DET(SI-F+G*C)..

-2.52094E+01:-1.04978E+00:-2.06591E-01:-5.02113E-02:

CONTROL EIGENVECTOR MATRIX.....C*M..

-3.326553E-C2	1.117587E-08	-6.622234E-04	-4.134316E-06
2.902164E+00	-1.370907E-06	-2.356589E-03	-5.652895E-06

*** THE STARTING/FINAL F MATRIX ***

0.00387	0.03151	0.00399	-0.06416
-0.24260	-2.83522	-0.34541	3.11642

*** THE AUGMENTED A-B*F*C MATRIX ***

-2.03999	4.10968	0.17766	29.03711
-2.02074	-24.43202	-2.96126	26.08698
0.04354	-0.00401	-0.04445	-0.04051
0.00000	1.00000	0.11000	0.00000

END OF CPTP ANALYSIS

[illegible]

*** THE R WEIGHTING MATRIX ***

$\frac{7}{9} = \frac{70000}{90000}$

*** THE STARTING/FINAL F MATRIX ***

[illegible]

*** THE ORDERED COMPLEX EIGENVALUES (INPUT)

-0.70000 1.42000

** ICOMP **

*** AUX TRANSFORMATION L ***

[illegible]

DIRECTOR
CF A

-0.13554
 -0.14603
 -0.01263
 0.00453

TRANSFORMED A MATRIX

-0.39354	-1.42648	-0.00001	0.00000
1.42646	-0.39355	0.00000	0.00000
0.00000	0.00001	0.13768	0.00000
0.00000	-0.00001	0.00000	-0.18058

TRANSFORMED Q MATRIX

*** TRANSFORMED B MATRIX ***

-24.68213
 -1.57558
 -11.17529
 6.63291
 -1.62757
 -1.52757
 -1.26164
 -1.73118
 -1.69931

*****SQUAPE RCCT INVERSE OF R *****

SQUARE	RCGT	INVERSE
0.70711		C.00000
0.00000		0.37796

[illegible]

0000
••••
0001

```

AAAAA DDDDD SSSSS
A A D D S
A A D D S
AAAAA D D SSSSS
A A D D S
A A D D S
A A D D S
A DDDDD SSSSS

```

F O R T R A N P R O G R A M
 F O R
 A U T O M A T E D D E S I G N S Y N T H E S I S
 V E R S I O N 1.00

```

CONTROL PARAMETERS
ISTRAT = 5 IOPT = 3 IONED = 3 IPRINT = 1000
IGRAD = 0 NDV = 2 NCCN = 3

```

1

 OPTIMIZATION RESULTS

OBJECTIVE FUNCTION VALUE 0.28365E-10

DESIGN VARIABLES

VARIABLE	LOWER BOUND	VALUE	UPPER BOUND
1	0.00000E+00	0.23088E-02	0.10000E+04
2	0.00000E+00	0.22134E-02	0.10000E+04

DESIGN CONSTRAINTS

1) -C.4999E-01 -C.9000E+00 0.9545E-04

FUNCTION EVALUATIONS = 48

```

*** THE OPTIMAL Q MATRIX
      0.00231      0.00000      0.00000      0.00000
      0.00000      0.00221      0.00000      0.00000
      0.00000      0.00000      0.00000      0.00000
      0.00000      0.00000      0.00000      0.00000

END OF POLE PLACEMENT ROUTINE

THE Q REQUIRED FOR THIS ASSIGNMENT IS

      0.00060      0.00106      -0.00908      0.00301
      0.00106      0.00189      -0.01970      0.00579
      -0.00908      -0.01970      12.94063      -1.60992
      0.00301      0.00579      -1.60992      0.20628

EIGENSYSTEM OF OPTIMAL REGULATOR.....

C-LQCP CFTIMAL REG. E-VALUES...DET(SI-F+G*C)..

-6.99887E-01, 1.41996E+00:-1.80579E-01:-1.37682E-01:

CONTROL EIGENVECTOR MATRIX.....C*M..

4.996814E-03 -2.528824E-02 -6.565824E-08 -5.181365E-03
2.956046E-04 -4.560151E-04 -3.259629E-09 -1.896438E-04

*** THE STARTING/FINAL F MATRIX ***
      -0.01311      -0.02106      2.67146      2.79805
      -0.00052      -0.00084      0.06034      0.09177

*** THE AUGMENTED A-B*F*C MATRIX ***
      -0.18440      -0.03706      10.48966      -30.92160
      -0.20052      -0.55084      109.49033      2.87177
      -0.00566      -0.01073      -0.98285      -0.92528
      0.00000      0.00000      1.00000      0.00000

END OF CPTPP ANALYSIS

```

[illegible]


```

**-----REIG - REAL COMPUTED EIGENVALUE OF THE SYSTEM-----**
**THE FOLLOWINGS ARE USED BY THE SINGULAR VALUE ANALYSIS MODE**
**-----**
OMEGA - FREQUENCY
SV - SINGULAR VALUE OF (I + F * PLANT TRANSFER FUNCTION(G))
SVM1 - SINGULAR VALUE OF (I + INV(F * G))
SVAC - SINGULAR VALUE OF (I + G * F)
SVMC - SINGULAR VALUE CF (I + INV(G * F))
SIGNM1 - ONLY OF INTEREST IF NO. OUTPUTS = NO. INPUTS.
SIGNM2 - FIRST SINGULAR VALUE CF (I + F * G)
SIGNMX - SECOND SINGULAR VALUE OF (I + F * G)
SIGPRG - MAXIMUM SINGULAR VAL * SECOND SING VAL (I + G * F)
SVADMO - SQRT(FIRST SING VAL * SINGULAR VALUE (I + INV(FG))
SVADMO - MIN ADDITIVE OUTPUT SINGULAR VALUE
SVMIX - MIN MULTIPLICATIVE INPUT SINGULAR VALUE (I + INV(FG))
SVMIX - MAX MULTIPLICATIVE INPUT SINGULAR VALUE
SVMC - MIN MULTIPLICATIVE OUTPUT SINGULAR VALUE (I + INV(GF))
SVMXO - MAX MULTIPLICATIVE CUTPUT SINGULAR VALUE
**-----**
IMPLICIT REAL*4(A-H,O-Z)
REAL*4 A(10,10),B(10,10),C(10,10),REALMU(50),IMAGMU(50),
1F(10,10),FC(10,10),BFC(10,10),W(10,10),WK(1000),MEIG(10),
2REIG(50),IEIG(50),SV(10),WM(10,10),WA(100),MINEIG(500),
3OMEGA(500),SIGNM1(500),SIGNM2(500),SIGNMX(500),SIGPRO(500),
4SVADMO(500),SVADXD(500),SVMIM(500),SVMIX(500),
7LP11(500),LP12(500),LP21(500),LP22(500),
5SVMMD(500),SVMXD(500),SVM1(10),SVAG(10),SVMQ(10),TEMPR(10,10),
6QW(10,10),RW(10,10),RWSP(10,10),RWSPL(10,10),RCOND,Z1(10)
**-----**
THE COMPLEX PARAMETERS USED IN THE PROGRAM
AX - COMPLEX A
BX - COMPLEX B
CX - COMPLEX C
FX - COMPLEX F
QWX - COMPLEX OF QW
RWX - COMPLEX OF RW
RWSQX - COMPLEX OF R*(-1/2)
**-----**
**THE FOLLOWINGS ARE USED IN THE COORDINATE TRANSFORMATION BLOCK**
**-----**
Z - EIGENVECTOR OF A ( M IN THE THESIS )

```

CON0C430

CON0C470
CON0C480
CON0C490
CON0C500
CON0C510
CON0C520
CON0C530
CON0C540
CON0C550
CON0C560
CON0C570
CON0C580
CON0C590
CON0C600
CON0C610
CON0C620
CON0C630
CON0C640

CON0C650
CON0C660
CON0C670
CON0C680
CON0C690
CON0C700

CON0C720
CON0C730
CON0C740
CON0C750
CON0C760
CON0C770

CON0C790

```

** ** ** ** **
CZI - INVERSE OF Z
CZTI - INVERSE OF TRANPOSE OF Z
LX - AUX TRANSFORMATION MATRIX FOR COMPLEX EIGENVALUE
LXI - INVERSE OF LX
ZLI - INVERSE OF (Z*LX)
** ** ** **
-- THE FOLLOWINGS ARE USED TO COMPUTE THE OBJECTIVE FUNCTION
--
GPS - COMPLEX MATRIX COMPUTE BY PLANT1 G(S)
      WHERE G(S)=C*(S1-A)*(-1) * B
      SAME AS ABOVE BUT G(-S)
      G(S)*R*(-1/2)
      GPSRSQ - G*GPSRSQ
      GGR - G*GGR
      RGGR - R*(-1/2) * GQGR
      XIRQM - I + RGQGR
      DETERM - AN ARRAY CONSISTING OF THE DET OF XIRQI
      TO BE USED IN THE OBJECTIVE FUNCTION OF THE OPTIMIZER
** ** ** **
-- THE FOLLOWING ARE USED BY THE SINGULAR VALUE ANALYSIS
--
EIG - EIGENVALUE OF A - B*F*C
MU - INPUT DESIRED POLE LOCATION
OMU - ORDERED VALUE OF MU
GEIG - ORDERED VALUE OF EIG
CZ - ORDERED EIGENVECTOR
XI - IDENTITY MATRIX
XXI - IDENTITY MATRIX
UI - INPUT ADDITIVE SINGULAR VECTOR ASSOCIATED WITH SV
VI - INPUT ADDITIVE SINGULAR VECTOR (SVAO)
UC - OUTPUT ADDITIVE SINGULAR VECTOR (SVAO)
VC - OUTPUT ADDITIVE SINGULAR VECTOR
UMI - INPUT MULTIPLICATIVE SINGULAR VECTOR
VMI - INPUT MULTIPLICATIVE SINGULAR VECTOR
UMO - OUTPUT MULTIPLICATIVE SINGULAR VECTOR
VMO - OUTPUT MULTIPLICATIVE SINGULAR VECTOR
FXPLT - F * F*G
XIPFX - I + TRANSFER FUNCTION (G)
PLAN - PLANT TRANSFER FUNCTION (G)
PLTFX - G*F
XIPPFX - I + G*F
PLTFXI - INV(F*G)
FXPLTI - INV(G*F)
XIFPLI - I + INV (F*G)
XIFPLFI - I + INV (F*F)
** ** ** **
CONOC780
CONOC800
CONOC810
CONOC820
CONOC830
CONOC840
CONOC850
CONOC860
CONOC870
CONOC880
CONOC890
CONOC900
CONOC910
CONOC920
CONOC930
CONOC940
CONOC950
CONOC960
CONOC970
CONOC980
CONOC990
CONO1C00
CONO1C10
CONO1C20
CONO1C30

```



```

C      READ (1,580) SVMINI,SVMINO,RJ,NIDG
C      W=WI9
C      NC=1
C      ----INPUT OPTIMIZER INPUT DATA (SEE ADS MANUAL)-----
C
C      READ (1,760) IGRAD,NDV,NCON,ISTRAT,IOPT,IONED,IPRINT,INFO
C      ----INPUT THE PARAMETER VALUES AND THE MATRICES A,B,C,F,QW,RW-----
C
C      READ (1,770) NROWA,NCOLA,NROWB,NCOLB,NROWC,NCOLC,NROWF,NCOLF,
C      1NROWQ,NCOLQ,NROWR,NCOLR,NMU
C      CALL REAC (A,NROWA,NCOLA)
C      CALL REAC (B,NROWB,NCOLB)
C      CALL REAC (C,NROWC,NCOLC)
C      CALL REAC (F,NROWF,NCOLF)
C      CALL REAC (QW,NROWQ,NCOLQ)
C      CALL REAC (RW,NROWR,NCOLR)
C      DO 13 I=1,NROWR
C      DO 13 J=1,NCOLR
C      RWSP(I,J)=RW(I,J)
C      CONTINUE
C
C      13 ----SPECIFY THE DESIGN VARIABLES WITHIN THE Q MATRIX-----
C      IF R OR BOTH Q AND R IS TO BE VARIED, MODIFICATION
C      IS REQUIRED FOR THE FOLLOWING BLOCK
C
C      DO 10 J=1,NROWQ
C      READ (1,780) (IQW(J,K),K=1,NCOLQ)
C      CONTINUE
C      10 ----SET THE DESIGN VARIABLE BOUNDS-----
C
C      DO 20 J=1,NDV
C      READ (1,790) VLB(J),VUB(J)
C      CONTINUE
C      20 ---- REAC THE DESIRED EIGENVALUES -----
C
C      DO 60 I=1,NMU
C      READ (1,860) REALMU(I),IMAGMU(I)
C      CONTINUE
C      60 DO 70 I=1,NMU
C      MU(I)=CMPLX(REALMU(I),IMAGMU(I))
C      CONTINUE
C      70 ---- SORT THE INPUT EIGENVALUES USING IMSL ROUTINE VSRTR-----
C      DO 80 I=1,NMU

```

```

CCN02350
CCN02360
CCN02370
CCN02380
CCN02390
CCN02400
CCN02410
CCN02420
CCN02430
CCN02440
CCN02450

CCN02460
CCN02470
CCN02480
CCN02490
CCN02460
CCN02470

CCN02500
CCN02510
CCN02520

CCN02530
CCN02540

CCN02550
CCN02560
CCN02570
CCN02580
CCN02590

CCN02710
CCN02720
CCN02730
CCN02740
CCN02750

CCN02760
CCN02770
CCN02780
CCN02800
CCN02810
CCN02820

```



```

C 190 CONTINUE CF DC LOOP -----
C -----WRITE (6,*) ,** ICCMP , *** , ICCMP
      WRITE (6,*) , *** AUX TRANSFORMATION L ***
      CALL CWRITE ( LX, NROWA, NCOLA)
      DO 200 J=1, NROWA
      KEY(J)=J
      CALL VSRTIP (REIG, NROWA, KEY)
      *** VSRTIP BY ALGEBRA OR ABS VALUE ***
      CALL VSRTIP (IEIG, NRCWA, KEY)
      DO 220 J=1, NCOLA
      K=KEY(NFCWA+1-J)
      DEIG(J)=IEIG(K)
      DO 210 KK=1, NROWA
      OZ (KK, J)=Z(KK, K)
      TEMPX(KK, J)=Z(KK, K)
      CONTINUE
      210 CONTINUE
      220 CONTINUE
C -----
C ANY EIGENVECTOR ASSIGNMENT ROUTINE MAY BE INERTED HERE IN PLACE OF
C THE EIGENVECTOR MATRIX OZ.
C -----
      WRITE (6,*) , *** EIGENVECTOR OF A ***
      CALL CPLREA (OZ, TEMPR, NROWA, NCOLA)
      CALL CWRITE (TEMPR, NROWA, NCOLA)
      DO 222 I=1, NROWA
      DO 222 J=1, NCOLA
      OZ I(I, J)=0.0
      OZ I(I, I)=1.0
      CONTINUE
      222 CALL CPLEQU (OZ I, OZTI, NROWA, NCOLA)
      CALL CPLEQU ( OZ, TEMPX2, NROWA, NCOLA)
      LEQ TIC DESTCRY OZ
      CALL LECTIC (TEMPIX, NROWA, 10, OZ I, NROWA, 10, O, WA, IER)
      CALL LECTIC (TEMPIX2, NROWA, 10, OZTI, NROWA, 10, O, WA, IER)
      CALL CPLXCV (AX, A, NROWA, NCOLA)
      CALL CPLXCV (BX, B, NROWB, NCLCB)
      CALL CPLXCV (QWX, QW, NROWQ, NCOLQ)
C -----
C ----- TRANSFORM AX (COMPLEX EIG VAULE ) -----
C IF (ICCMP.EQ.1) THEN

```

CCN03800

CCN03810
CCN03820
CCN03830CCNC3840
CCN03850
CCN03860
CCN03870
CCN03880CCN03890
CCNC3500CON03590
CCN03590
CONC3590

```

C      CALL CMATML (OZ,LX,NROWA,NCOLA,NROWA,TEMPX2)
C      CALL CMATML (AX,TEMPX2,NROWA,NCOLA,NROWA,TEMPX)
C      CALL CMATML (OZI,TEMPX,NROWA,NCOLA,NROWA,TEMPX2)
C      CALL CMATML (LXI,TEMPX2,NROWA,NCOLA,NROWA,AX)
C
C----- TRANSFORM BW -----
C
C      CALL CPLEQU (BX,TEMPX,NROWB,NCOLB)
C      CALL CMATML (OZ,LX,NROWA,NCOLA,NROWA,TEMPX2)
C      CALL LECTIC (TEMPX2,NROWA,IO,ZLI,NROWA,IO,O,WA,IER)
C      CALL CMATML (ZLI,TEMPX,NROWA,NROWC,NCOLB,BX)
C
C----- TRANSFORM GWX -----
C
C      CALL CMATML (OZ,LX,NROWA,NCOLA,NROWA,TEMPX2)
C      CALL CMATML (QWX,TEMPX2,NROWQ,NCOLQ,NROWA,TEMPX)
C      CALL CMATMT (TEMPX2,TEMPX,NROWC,NCOLQ,NROWA,QWX)
C
C      ELSE
C
C----- TRANSFORM AX (REAL EIG VALUE)-----
C
C      CALL CMATML (AX,OZ,NROWA,NCOLA,NROWA,TEMPX)
C      CALL CMATML (OZI,TEMPX,NROWA,NCOLA,NROWA,AX)
C
C----- TRANSFORM BX -----
C
C      CALL CPLEQU (BX,TEMPX,NROWB,NCOLB)
C      CALL CMATML (OZI,TEMPX,NROWA,NROWC,NCOLB,BX)
C
C----- TRANSFORM QWX -----
C
C      CALL CMATML (QWX,OZ,NROWQ,NCOLQ,NROWA,TEMPX)
C      CALL CMATMT (OZ,TEMPX,NROWQ,NCOLQ,NROWA,QWX)
C      END IF
C
C----- END OF IF THEN ELSE BLOCK -----
C
C----- CUTPUT TRANSFORMED A,B,Q -----
C      WRITE (6,*) ' *** TRANSFORMED A MATRIX *** '
C      CALL CPLREA (AX,TEMPR,NROWA,NCOLA)
C      CALL WRITE (TEMPR,NROWA,NCOLA)
C      WRITE (6,*) ' *** TRANSFORMED Q MATRIX *** '
C      CALL CPLREA (QWX,TEMPR,NROWQ,NCOLQ)
C      CALL WRITE (TEMPR,NROWQ,NCOLQ)
C      WRITE (6,*) ' *** TRANSFORMED B MATRIX *** '
C      CALL CPLREA (BX,TEMPR,NROWB,NCOLB)
C      CALL WRITE (TEMPR,NROWB,NCOLB)

```

```

C----- END OF TRANSFORMATION BLOCK -----
C
C----- FORMULATE THE DESIGN VARIABLE X FROM THE STARTING Q -----
C
      DO 50 J=1,NROWQ
      DO 40 K=1,NCULQ
      IF (IQW(J,K).EQ.0) GO TO 30
      KK=IQW(J,K)
      X(KK)=REAL(CWX(J,K))
      CONTINUE
    30 CONTINUE
    40 CONTINUE
    50 CONTINUE
C-----
      CALL CPLXCV (RWX,RW,NROWR,NCOLR)
C-----
C----- CALCULATE R**1/2 -----
C----- USING LINPACK ROUTINE AND WURZEL ROUTINE -----
C
      CALL SPCCO(RW,IO,NCCLR,RCOND,Z1,INFCL)
      CALL SPCCI(RW,IO,NCCLR,DET,1)
      DO 131 I=1,NROWR
      DO 131 J=1,NROWR
      RW(J,I)=RW(I,J)
      131 CONTINUE
C----- COMPUTE SQUARE ROOT OF MATRIX R -----
      CALL WURZEL(RW,TEMPR,NROWR,0.99,1.E-6,10)
      CALL CPLXCV (RWSQX,TEMPR,NROWR,NCOLR)
      WRITE (6,*) '***** SQUARE ROOT INVERSE OF R ***'
      CALL WRITE(TEMPR,NROWR,NROWR)
C-----
C----- ENTER OPTIMIZATION BLOCK HERE*****
C
      IF KGNTRL SET GREATER THAN 0 THE FIRST PASS JUMPS THE OPTIMIZER
      IF (KCNTRL.GT.0) GO TO 130
C-----
      CALL THE ADS OPTIMIZER TO SOLVE FOR THE BEST Q'S
C-----
C-----
      THIS FIRST CALL SETS THE SCALE FLAG SEE ADS MANUAL
      INFO=-2
      CALL ACS (INFO,ISTRAT,IOPT,IONED,IPRINT,IGRAD,NDV,NCON,X,VLB,VUB,
      1BJ,G,ICG,NGT,IC,DF,XA,NRA,NCA,WK1,NRWK,IWK,NRIWK)

```

CCN02610
CCN02620
CCN02630
CCN02640
CCN02650
CCN02660
CCN02670
CCN02680
CCN02690
CCN02700
CCN03600

CCN03590

CCN03170
CCN03180
CCN03190
CCN03200
CCN03210
CCN03220
CCN03230
CCN03240
CCN03250
CCN03260
CCN03270
CCN03280
CCN03290
CCN03300

```

C      WK(3)=-C.05
C      IWK(2)=0      IMPLIES NO SCALING
C      IWK(2)=0
C
C      THIS CALL STARTS THE OPTIMIZATION PROCESS
C
C120  CALL ACS (INFO,ISTRAT,IOPT,IONED,IPRINT,IGRAD,NDV,NCON,X,VLE,VUB,
      1BJ,G,IEG,NGT,IC,DF,XA,NRA,NCA,WKI,NRWK,IWK,NRIWK)
C      IF (INFC.EQ.0) GO TO 480
C      IF (INFC.GT.1) GO TO 470
C
C      THIS PORTION OF THE PROGRAM DOES THE ACTUAL COMPUTATION
C
C130  DO 160 J=1,NROWQ
C      DO 150 K=1,NCOLQ
C
C      ---EQUATE DESIGN VECTOR X WITH THE DESIRED SYSTEM MATRIX---
C
C      IF (IQW(J,K).EQ.0) GO TO 140
C      KK=IQW(J,K)
C      QWX(K,J)=X(KK)
C      QWX(J,K)=X(KK)
C      CONTINUE
C      CONTINUE
C      CONTINUE
C
C      CALL CPLXCV {CX,C,NRCWC,NCOLC}
C      CALL CPLXCV {FX,F,NROWF,NCCLF}
C
C      MAIN LCOP TO COMPUTE THE DET (1+R**(-1/2))G(-S)**T *Q *G(S)R**(-1/2))
C      TO BE USED IN THE OPTIMIZER'S OBJECTIVE FUNCTION
C
C      DO 263 I=1,NMU
C
C      --- PLANT1 COMPUTE G(S) WHERE G=C(SI-A)**(-1)*B---
C      WHERE S IS THE DESIRED POLE LCCATION
C
C      CALL PLANT1(GPS,AX,BX,CX,MU(I),NROWA,NCOLA,NROWB,NCOLB,NROWC,
C      1NCCLC)
C
C      --- PLANT1 COMPUTE G(-S) WHERE G=C(SI-A)**(-1)*B---
C      WHERE S IS THE DESIRED POLE LCCATION

```

```

      CALL PLANTI(GMS,AX,BX,CX,-MU(1),NRDWA,NCOLA,NROWB,NCOLB,NROWC,
1 NCGLC)
      CALL CMATML (GPS,RMSGX,NROWA,NCOLA,NCOLR,GPSRSQ)
      CALL CMATML (QMX,GPSFSQ,NROWA,NCOLA,NCOLR,QGR)
      CALL CMATML (GMS,QGR,NROWA,NCOLA,NCOLR,GQGR)
      CALL CMATML (RMSGX,GQGR,NROWA,NCOLA,NCOLR,RGQGR)
-----
      I+R**-(1/2)IG(-S)**TIG(S)IR**-(1/2)I
-----
      DO 911 J=1,NROWR
      DO 913 K=1,NCOLR
      XXI(J,K)=0.0
      XXI(J,J)=1.0
      XIQR(J,K)=XXI(J,K)+RGQGR(J,K)
913 CONTINUE
911 CONTINUE
-----
      FIND DETERMINANT TO BE USED IN THE OBJECTIVE FUNCTION ---
      USING LINPACK ROUTINE
-----
      CALL CGECC(XIQR,10,NCOLR,IPVT,PCOND,WORK)
      CALL CGEDI(XIQR,10,NCOLR,IPVT,DET,WORK,10)
      DETERM(1)=DET(1)*10.**REAL(DET(2))
-----
253 CONTINUE
-----
      END OF DG LOOP
-----
      C--FORM QUANTITIES TO USE IN THE COST CRITERIA OF OPTIMIZER-----
      C EACH REAL POLE REQUIRED ONE COST, EACH COMPLEX PAIR POLE REQUIRED
      C TWO COST(COST1 AND COST2)
      COST1= ((REAL(DETERM(1)))**2+(AIMAG(DETERM(1)))**2)
      COST2= ((REAL(DETERM(2)))**2+(AIMAG(DETERM(2)))**2)
      COST3= ((REAL(DETERM(3)))**2+(AIMAG(DETERM(3)))**2)
      WRITE(6,*) 'X',COST1,DETERM(1),X(1)
-----
      C--THE OBJECTIVE FUNCTION IS INSERTED HERE-----
      OBJ= COST1+COST2+COST1*COST2
-----
      WRITE(6,*) 'OBJ'
      C--CONSTRAINT EQUATION (IDG(3))=-2 FOR COMPLEX ROOTS)-----
      IF (NCCN.EQ.0) GO TO 460
      DO 450 J=1,2
      IDG(J)=NIDG
      CONTINUE
450 IDG(3)=-2
      C-----OTHER CONSTRAINTS CAN BE INTRODUCED HERE-----

```

CCN04240
CCN04240
CCN04240
CCN04240

CCN05000
CCN05010

CCN05130
CCN05140

CCN05180
CCN05190
CCN05200
CCN05210

[illegible]

CCND 5240
CCND 5240

```

WRITE (6,*) '
CALL WRITE (Q,NROWC,NCOLC)
-----SINGULAR VALUE ANALYSIS ONLY -----
531 IF (KCODE.EQ.2.AND.KCNTRL.EQ.2) GO TO 532
N2=2*NCOLA
*****
----- THE REDUCE ROUTINE MATCHES THE VARIABLE FROM THE OPTPP
WITH THE MODIFIED VERSION OF THE INNER ROUTINE FROM
CPTSYS PROGRAM
-----

THE SYSTEM MATRIX A,B,C, AND Q, F(FROM CPTPP) ARE USED
BY THE INNER ROUTINE TO COMPUTE THE FEEDBACK GAIN F.
F CAN THEN BE USED IN THE SINGULAR VALUE ANALYSIS PORTION OF
THIS PROGRAM
-----

CALL REDUCE(A,B,C,Q,RWSP,DUMA,DUMB,DUMC,DUMQ,DUMR,DUMRI,
INRCKA,NCCLB)
-----
CALL INNER (NCOLA,NCCLB,NRCWC,N2,DUMA,DUMB,DUMC,DUMQ,DUMR,FBGC
1,KM,AA,PRG,XI,GM,DUMRI)
DO 533 I=1,NROWF
DO 533 J=1,NCOLF
F(I,J)=-FBGC(I,J)
533 CONTINUE
532 CALL CPLXCV (AX,A,NROWA,NCOLA)
CALL CPLXCV (BX,B,NROWB,NCCLB)
CALL CPLXCV (CX,C,NRCWC,NCCLC)
CALL CPLXCV (FX,F,NROWF,NCOLF)
----- THE FINAL FEEDBACK GAIN MATRIX IS DISPLAY HERE-----
WRITE (6,840)
CALL WRITE (F,NROWF,NCOLF)
-----
COMPUTE THE A-B*F*C SYSTEM MATRIX FOR THE NEXT PGLE ASSIGNMENT
-----
CALL MMUL (F,C,NROWF,NCOLF,NCOLC,FC)
CALL MMUL (B,FC,NRCWB,NCOLE,NCOLC,BFC)
DO 180 I=1,NROWA
DO 170 J=1,NCOLA

```

CCN03590
CCN03590
CCN03590

CCN03130
CCN03140

CCN03630
CCN03640
CCN03650

CCN03660
CCN03670
CCN03680
CCN03690

```

170 AMBFC(I,J)=A(I,J)-BFC(I,J)
180 CONTINUE
C WRITE (C,890)
C -----OUTPUT (A-BFC) -----
C CALL WRITE (AMBFC,NROWA,NCOLA)
C
C ***** GO TO 555 *****
C SINGULAR VALUE ANALYSIS BEGIN HERE
C -----
C CALL THE PLANT TRANSFER MATRIX ROUTINE AND FORM THE SYSTEM
C RETURN DIFFERENCE MATRICES AS REQUIRED
C -----
DEL=DELW
*II=W19
NCNT=1
W=W19
SMINAI=C.
SMINMI=C.
SMINAO=0.
SMINMO=C.
SMAXAI=C.
SMAXMI=C.
SMAXAO=C.
SMAXMO=C.
CALL CPLXCV (AX,A,NROWA,NCOLA)
CALL CPLXCV (BX,B,NROWB,NCOLB)
CALL CPLXCV (CX,C,NROWC,NCOLC)
CALL PLANT (PLAN,AX,BX,CX,NROWA,NCOLA,NROWB,NCOLB,NROWC,NCOLC)
CALL CMATML (FX,PLAN,NROWF,NCOLF,NCOLB,NCCLF,PLTFX)
DO 280 J=1,NROWF
DO 270 K=1,NCOLB
XI(J,K)=0.0
CONTINUE
DO 290 J=1,NROWF
XI(J,J)=1.0
DO 310 J=1,NROWC
DO 300 K=1,NCOLF
XXI(J,K)=0.0
XXI(J,J)=1.0
XI*PFX(J,K)=XXI(J,K)+PLTFX(J,K)
CONTINUE
DO 330 J=1,NROWF

```

CCN03700
CCN03710
CCN03530

CCN05540
CCN05550

CCN04C50
CCN04C60
CCN04C70
CCN04C80
CCN04C90
CCN04100
CCN04110
CCN04120
CCN04130
CCN04140
CCN04150
CCN04160
CCN04170
CCN04180
CCN04190
CCN04200
CCN04210
CCN04220
CCN03590
CCN03590
CCN04230
CCN04240
CCN04250
CCN04260
CCN04270
CCN04280
CCN04290
CCN04300
CCN04310
CCN04320
CCN04330
CCN04340
CCN04350
CCN04360
CCN04370
CCN04380
CCN04390

```

320 DO 320 K=1,NCOLB
330 XIPEXP(J,K)=XI(J,K)+FXPLT(J,K)
340 CALL CPLEQU (XIPEXP,TEMPX,NROWF,NROWF)
350 DO 350 J=1,NROWF
360 DO 360 K=1,NROWF
370 FXPLTI(J,K)=0.0
380 FXPLTI(J,J)=1.0
390 CONTINUE
400 DO 400 J=1,NROWC
410 DO 400 K=1,NROWC
420 PLTFXI(J,K)=J.0
430 PLTFXI(J,J)=1.0
440 CONTINUE
450 C THIS ESTABLISHES THE RETURN DIFFERENCE MATRIX FOR MULT. CASE
460 C
470 C
480 CALL LECTIC (FXPLT,NROWF,10,FXPLTI,NROWF,10,0,WA,IER)
490 CALL LECTIC (PLTFX,NROWC,10,PLTFXI,NROWC,10,0,WA,IER)
500 DO 500 J=1,NROWF
510 DO 500 K=1,NCOLB
520 XIPLI(J,K)=XI(J,K)+FXPLTI(J,K)
530 CONTINUE
540 CONTINUE
550 DO 550 J=1,NROWC
560 DO 550 K=1,NROWC
570 XIPLFI(J,K)=XI(J,K)+PLTFXI(J,K)
580 CONTINUE
590 CONTINUE
600 -----
610 DO SINGULAR VALUE DECCMP AND QUANTIFY ALL DESIRED SV'S
620 -----
630 CALL CSVC (XIPEXP,10,10,NROWF,NROWF,0,NROWF,NROWF,SV,UI,VI)
640 CALL CSVD (XIPEXP,10,10,NROWC,NROWC,0,NROWC,NROWC,SVAG,LO,VO)
650 CALL CSVD (XIPLI,10,10,NROWF,NROWF,0,NROWF,NROWF,SVMI,UMI,VMI)
660 CALL CSVD (XIPLFI,10,10,NROWC,NROWC,0,NROWC,NROWC,SVMO,LMO,VMO)
670 -----
680 C----- THIS PORTION COMPUTES THE MAXIMUM EIG-VALUES OF THE -----
690 C RETURN DIFFERENCE MATRIX WHICH GIVE THE UPPER BOUND
700 C FOR THE SINGULAR VALUE
710 C
720 CALL EIGCC(TEMPX,NROWF,10,0,EIG,Z,10,WK,IER)
730 DO 730 I=1,NROWF
740 MEIG(I)=SQRT((REAL(EIG(I))**2+(AIMAG(EIG(I))**2)
750 IF (I.EC.1) THEN
760 MEIG(I)=MEIG(I)

```



```

C----- IF (W.LE.WMAX) GO TO 260
C          MAXN=NCNT-1
C----- OUTPUT THE OPTIMIZED SINGULAR VALUE DATA-----
C
C          WRITE (6,73C) MAXN
C          WRITE (3,66C) MAXN
C          DO 540 J=1,MAXN
C            WRITE (3,67C) OMEGA(J),SIGNM1(J),SIGNM2(J),SIGNMX(J),SIGPRO(J)
C            WRITE (3,74C) OMEGA(J),SIGNM1(J),MINEIG(J),SIGNMX(J),SIGPRE(J)
C            WRITE (6,68C)
C          DO 550 J=1,MAXN
C            WRITE (3,69C) SVADM(J),SVADXD(J),SVMIM(J),SVMIX(J),SVMMD(J),SVMXO
C            I(J)
C          WRITE (3,74C) OMEGA(J),SIGNM1(J),LP11(J),LP12(J),LP22(J)
C          WRITE (6,70C) OMEGA(J),SVADM(J),SVADXD(J),SVMIM(J),SVMIX(J),SVMMD
C          I(J),SVMXO(J)
C          IF (KONTRL.EQ.0.OR.KONTRL.GT.1) GO TO 560
C          WRITE (6,*) 'KONTRL',KONTRL
C          KONTRL=C
C          GO TO 12C
C          WRITE (6,*) 'END OF OPTPP ANALYSIS '
C          STOP
C----- *****
C----- FCRMAT STATEMENTS
C-----
C          FORMAT (3F10.0)
C          FORMAT (3F10.0,15)
C          FCRMAT (/,10H WEIGHT 1=,F10.5,9HWEIGHT 2=,F10.5,9HWEIGHT 3=,F10.5,
C          1/)
C          FCRMAT (/,9H SVMINI= ,F10.5,8H SVMINO=,F10.5,5H RJ=,F10.5,6H IDG
C          1=,15,/)
C          FCRMAT (215)
C          FCRMAT (/,28H *** THE INITIAL DATA IS ***,/)
C          FCRMAT (/,41H *** THE OPTIMIZER COMPUTED OUTPUT IS ***,/)
C          FCRMAT (2E12.5)
C          FCRMAT (2E12.5)
C          FCRMAT (/,34H *** THE OBJECTIVE FUNCTION IS =,F10.5,6H ***,/)
C          FCRMAT (15)
C          FCRMAT (5E12.5)
C          FCRMAT (/,2X,7H FREQ ,5X,6HSVADM0,4X,6HSVADXD,4X,5HSVMIM,4X,5HSVM
C          1IX,4X,5HSVMMD,4X,5HSVMXO,/)
C          FCRMAT (6E12.5)
C          FCRMAT (7F10.5)
C          FCRMAT (3F10.0)
C          FCRMAT (/,6H WMAX=,F10.5,1X,10HINIT FREQ=,F10.5,1X,10HFREQ STEP=,F
C          10.5,/)
C          FCRMAT (/,10H FREQUENCY,5X,13HMIN ADD IN SV,4X,13H2ND ADD IN SV,3C
C          730

```

740
750
760
770
780
790
800
810
820
830
831
832
840
841
850
860
870
880
890
900

137


```

370 DO 370 K=1,NC
    PRG(I,J)=PRG(I,J)+BLI(I,K)*GL(J,K)
CONTINUE
380 DO 380 I=1,MH
    DO 380 J=1,MH
        RM(I,J+MH)=C.O
    DO 380 K=1,NC
        RM(I,J+MH)=(RM(I,J+MH)-GL(I,K)*PRG(K,J))
CONTINUE
380 C-----2NX2N HAMILTONIAN MATRIX-----
C-----DIAGONAL BLOCKS-----M11 AND M22-----
C
C      WRITE (6,*) 'BA'
C      CALL WRITE (BA,2,2)
C      DO 390 I=1,MH
C      DO 390 J=1,MH
C      RM(I,J)=C.EC
C      RM(I,J)=BA(I,J)
C      RM(I+MH,J+MH)=-BA(J,I)
C-----M21 BLOCK-----
390 RM(I+MH,J)=-RM(I+MH,J)
CONTINUE
400 C-----M12 BLOCK IS DEFINED IN LINE 430 ABOVE-----
C-----
410 C-----CALL EISPACK ROUTINES-----
C      CALL BALANC (M,M,LM,LOW,HIGH,D1)
C      CALL ORTHAN (M,M,LM,LOW,HIGH,DM,D2)
C      CALL CRTRAN (M,M,LM,LOW,HIGH,RM,DZ,X1)
C      CALL HGR2 (M,M,LM,LOW,HIGH,RM,WR,X1,X1,IERR)
C      WRITE (6,*) IERR, IERR
C      IF (IERR.NE.0) CALL EREXIT (M,RM,IERR)
C      CALL BALBAK (M,M,LOW,HIGH,D1,M,X1)
C-----DEBUG DIAGNOSTICS ON EULER-LAGRANGE EQUATIONS-----
450 CONTINUE
IF (NUP.EQ.0) WRITE (6,1550)
C-----CALL RGAIN TO COMPUTE THE FEEDBACK GAIN-----
C      CALL RGAIN (M,NS,NC,NCB,WR,WI,X1,GN,W1,RM,W2,D1,CWR,CWI,SC,MHS,
1D2)
450 CONTINUE
470 CONTINUE
C-----CALCULATION OF FEEDBACK GAIN-----
C-----FEEDBACK GAINS--> U = -(BINVERSE)*GT*G1&-----
C-----CALCULATE GT-----
C      DO 480 I=1,NS
C      DO 480 J=1,NS
C      PRG(I,J)=C.O
C      DO 480 K=1,NS
C      PRG(I,J)=PRG(I,J)+GL(K,I)*GN(K,J)
C      DO 480 I=1,NC

```

CPT08560
CPT08570
CPT08580
CPT08590
CPT08500

CPT08510
CPT08520

CPT08530
CPT08540

CPT08550
CPT08560
CPT08570
CPT08580
CPT08590
CPT08500

CPT08540
CPT08550
CPT08560
CPT08570
CPT08580
CPT08590
CPT08500

CPT08580
CPT08590
CPT08500
CPT08510
CPT08520
CPT08530
CPT08540
CPT08550
CPT08560
CPT08570
CPT08580
CPT08590
CPT08500

```

CPT05290
CPT05400
CPT05410
CPT05420

D) 490 J=1,NS
FBGC(I,J)=0.0
DO 490 K=1,NC
  FBGC(I,J)=FBGC(I,J)-BLI(I,K)*PRO(K,J)
CONTINUE
490
C-----NORMALIZE AND PRINT OPT. REG. CLOSED LOOP EIGENSYSTEM-----
  IWRITE=2
  CALL CNCRM (CWR,CWI,SC,NS,IWRITE,NSC,DDC,D1,D2,WNORM,WNCRMI,FBGC,
    1AA,NC,NS)
C-----THE OPTIMUM FEEDBACK CONTROL GAINS-----
500 CONTINUE
RETURN
C-----
1380 FORMAT (//,5X,45HOPFN LOOP DYNAMICS MATRIX.....F.,//)
1390 FORMAT (10(2X,OPD11.4))
1400 FORMAT (//,5X,45HTHE CONTROL DISTRIBUTION MATRIX.....G.,//)
1410 FORMAT (//,5X,45HTHE CONTROL COST MATRIX.....B.,//)
1420 FORMAT (//,5X,45HMEASUREMENT SCALING MATRIX.....F.,//)
1430 FORMAT (//,5X,45HOUTPUT COST MATRIX.....A.,//)
1440 FORMAT (//,5X,45HDETERMINANT CASE.....//,10X,39HTHE EIGENVALUES.....)
1450 FORMAT (//,25X,28H.....DETERMINANT CASE.....//,10X,39HTHE EIGENVALUES.....)
1460 FORMAT (//,25X,28H.....DETERMINANT CASE.....//,10X,39HTHE EIGENVALUES.....)
1470 FORMAT (//,25X,28H.....DETERMINANT CASE.....//,10X,39HTHE EIGENVALUES.....)
1480 FORMAT (//,25X,28H.....DETERMINANT CASE.....//,10X,39HTHE EIGENVALUES.....)
1490 FORMAT (//,25X,28H.....DETERMINANT CASE.....//,10X,39HTHE EIGENVALUES.....)
1500 FORMAT (//,25X,28H.....DETERMINANT CASE.....//,10X,39HTHE EIGENVALUES.....)
1510 FORMAT (//,25X,28H.....DETERMINANT CASE.....//,10X,39HTHE EIGENVALUES.....)
1520 FORMAT (//,25X,28H.....DETERMINANT CASE.....//,10X,39HTHE EIGENVALUES.....)
1530 FORMAT (//,25X,28H.....DETERMINANT CASE.....//,10X,39HTHE EIGENVALUES.....)
1540 FORMAT (//,25X,28H.....DETERMINANT CASE.....//,10X,39HTHE EIGENVALUES.....)
1550 FORMAT (//,25X,28H.....DETERMINANT CASE.....//,10X,39HTHE EIGENVALUES.....)
1560 FORMAT (//,25X,28H.....DETERMINANT CASE.....//,10X,39HTHE EIGENVALUES.....)
1570 FORMAT (//,25X,28H.....DETERMINANT CASE.....//,10X,39HTHE EIGENVALUES.....)
1580 FORMAT (//,25X,28H.....DETERMINANT CASE.....//,10X,39HTHE EIGENVALUES.....)
1590 FORMAT (//,25X,28H.....DETERMINANT CASE.....//,10X,39HTHE EIGENVALUES.....)
1600 FORMAT (//,25X,28H.....DETERMINANT CASE.....//,10X,39HTHE EIGENVALUES.....)
1610 FORMAT (//,25X,28H.....DETERMINANT CASE.....//,10X,39HTHE EIGENVALUES.....)
1620 FORMAT (//,25X,28H.....DETERMINANT CASE.....//,10X,39HTHE EIGENVALUES.....)
1630 FORMAT (//,25X,28H.....DETERMINANT CASE.....//,10X,39HTHE EIGENVALUES.....)
1640 FORMAT (//,25X,28H.....DETERMINANT CASE.....//,10X,39HTHE EIGENVALUES.....)
1650 FORMAT (//,25X,28H.....DETERMINANT CASE.....//,10X,39HTHE EIGENVALUES.....)
1660 FORMAT (//,25X,28H.....DETERMINANT CASE.....//,10X,39HTHE EIGENVALUES.....)
1670 FORMAT (//,25X,28H.....DETERMINANT CASE.....//,10X,39HTHE EIGENVALUES.....)
1680 FORMAT (//,25X,28H.....DETERMINANT CASE.....//,10X,39HTHE EIGENVALUES.....)
1690 FORMAT (//,25X,28H.....DETERMINANT CASE.....//,10X,39HTHE EIGENVALUES.....)
1700 FORMAT (//,25X,28H.....DETERMINANT CASE.....//,10X,39HTHE EIGENVALUES.....)
1710 FORMAT (//,25X,28H.....DETERMINANT CASE.....//,10X,39HTHE EIGENVALUES.....)
1720 FORMAT (//,25X,28H.....DETERMINANT CASE.....//,10X,39HTHE EIGENVALUES.....)
1730 FORMAT (//,25X,28H.....DETERMINANT CASE.....//,10X,39HTHE EIGENVALUES.....)
1740 FORMAT (//,25X,28H.....DETERMINANT CASE.....//,10X,39HTHE EIGENVALUES.....)
1750 FORMAT (//,25X,28H.....DETERMINANT CASE.....//,10X,39HTHE EIGENVALUES.....)
1760 FORMAT (//,25X,28H.....DETERMINANT CASE.....//,10X,39HTHE EIGENVALUES.....)
1770 FORMAT (//,25X,28H.....DETERMINANT CASE.....//,10X,39HTHE EIGENVALUES.....)
1780 FORMAT (//,25X,28H.....DETERMINANT CASE.....//,10X,39HTHE EIGENVALUES.....)
1790 FORMAT (//,25X,28H.....DETERMINANT CASE.....//,10X,39HTHE EIGENVALUES.....)
1800 FORMAT (//,25X,28H.....DETERMINANT CASE.....//,10X,39HTHE EIGENVALUES.....)
1810 FORMAT (//,25X,28H.....DETERMINANT CASE.....//,10X,39HTHE EIGENVALUES.....)
1820 FORMAT (//,25X,28H.....DETERMINANT CASE.....//,10X,39HTHE EIGENVALUES.....)
1830 FORMAT (//,25X,28H.....DETERMINANT CASE.....//,10X,39HTHE EIGENVALUES.....)
1840 FORMAT (//,25X,28H.....DETERMINANT CASE.....//,10X,39HTHE EIGENVALUES.....)
1850 FORMAT (//,25X,28H.....DETERMINANT CASE.....//,10X,39HTHE EIGENVALUES.....)
1860 FORMAT (//,25X,28H.....DETERMINANT CASE.....//,10X,39HTHE EIGENVALUES.....)
1870 FORMAT (//,25X,28H.....DETERMINANT CASE.....//,10X,39HTHE EIGENVALUES.....)
1880 FORMAT (//,25X,28H.....DETERMINANT CASE.....//,10X,39HTHE EIGENVALUES.....)
1890 FORMAT (//,25X,28H.....DETERMINANT CASE.....//,10X,39HTHE EIGENVALUES.....)
1900 FORMAT (//,25X,28H.....DETERMINANT CASE.....//,10X,39HTHE EIGENVALUES.....)
1910 FORMAT (//,25X,28H.....DETERMINANT CASE.....//,10X,39HTHE EIGENVALUES.....)
1920 FORMAT (//,25X,28H.....DETERMINANT CASE.....//,10X,39HTHE EIGENVALUES.....)
1930 FORMAT (//,25X,28H.....DETERMINANT CASE.....//,10X,39HTHE EIGENVALUES.....)
1940 FORMAT (//,25X,28H.....DETERMINANT CASE.....//,10X,39HTHE EIGENVALUES.....)
1950 FORMAT (//,25X,28H.....DETERMINANT CASE.....//,10X,39HTHE EIGENVALUES.....)
1960 FORMAT (//,25X,28H.....DETERMINANT CASE.....//,10X,39HTHE EIGENVALUES.....)
1970 FORMAT (//,25X,28H.....DETERMINANT CASE.....//,10X,39HTHE EIGENVALUES.....)
1980 FORMAT (//,25X,28H.....DETERMINANT CASE.....//,10X,39HTHE EIGENVALUES.....)
1990 FORMAT (//,25X,28H.....DETERMINANT CASE.....//,10X,39HTHE EIGENVALUES.....)
2000 FORMAT (//,25X,28H.....DETERMINANT CASE.....//,10X,39HTHE EIGENVALUES.....)
END

```

```

C-----
C  REDUCE 10*10 MATRIX TO MATCH VARIABLE BET. OPTPP AND INNER
C-----
C  SUBROUTINE REDUCE (A,B,C,Q,P,DJMA,DUMB,DUMQ,DUMR,DUMRI,NS,
C  INC)
C  REAL*4 A(10,10),B(10,10),C(10,10),F(10,10),G(10,10),R(10,10)
C  REAL*4 CUMA(NS,NS),CUMB(NS,NS),DUMC(NS,NS),
C  DUMQ(NS,NS),DUMR(NC,NC),DUMRI(NC,NC)
C  INTEGER NS,NO,NC
C  DO 10 I=1,NS
C  DO 10 J=1,NS
C  DUMA(I,J)=A(I,J)
C  DUMB(I,J)=B(I,J)
C  DUMC(I,J)=C(I,J)
C  DUMQ(I,J)=Q(I,J)
C  DUMR(I,J)=R(I,J)
C  CONTINUE
C  DO 30 I=1,NC
C  DO 30 J=1,NC
C  DUMRI(I,J)=R(I,J)
C  CONTINUE
C  RETURN
C  END
C-----
C  WRITE THE COMPLEX MATRIX
C-----
C  SUBROUTINE CWRITE (A,N,K)
C  COMPLEX *8A(10,10)
C  INTEGER N,K,I,J
C  DO 10 I=1,N
C  WRITE (6,20) (A(I,J),J=1,K)
C  CONTINUE
C  RETURN
C  FORMAT (8F12.5)
C  END
C-----
C  WRITE THE COMPLEX VECTOR
C-----
C  SUBROUTINE CVECWR (A,N)
C  COMPLEX *8A(10)
C-----
C  WRITE THE COMPLEX VECTOR
C-----
C  SUBROUTINE CWRITE (A,N,K)
C  COMPLEX *8A(10,10)
C  INTEGER N,K,I,J
C  DO 10 I=1,N
C  WRITE (6,20) (A(I,J),J=1,K)
C  CONTINUE
C  RETURN
C  FORMAT (8F12.5)
C  END
C-----
C  WRITE THE COMPLEX VECTOR
C-----
C  SUBROUTINE CVECWR (A,N)
C  COMPLEX *8A(10)
C-----

```

CWRC220
 CWRC230
 CWRC240
 CWRC250
 CWRC260
 CWRC270

```

INTEGER N,J
WRITE(C,10) (A(J),J=1,N)
RETURN
FORMAT (2F12.5)
END

```

[illegible]

CWR0C280

WRITE THE REAL VECTOR

001020304050607080
001020304050607080
0000000000000000
0000000000000000
R R R R R R R R R R
C C C C C C C C C C

```

SUBROUTINE VECWR (A,N,IR)
  REAL*4 A(10)
  DIMENSION IR(50)
  INTEGER N,J
  WRITE(6,10) (A(J),IR(J),J=1,N)
  RETURN
  FORMAT (F12.5,5X,I5)

```

[illegible]

CWR0C390

MATRIX MULTIPLICATION WITH COMPLEX MATRIX

[illegible]

```

SUBROUTINE CMATML (R,T,M,LL,N,U)
  COMPLEX *8 R(10,10),T(10,10),J(10,10)
  INTEGER X,LL,N
  DO 20 I=1,M
    DO 20 J=1,N
      U(I,J)=C(1,J)+R(I,INDEX)*T(INDEX,J)
    DO 10 INDEX=1,LL
  
```

[illegible]

CWR00550

REAL MATRIX MULTIPLICATION ROUTINE

CWROCE570
 CWROCE580
 CWROCE590
 CWROCE600
 CWROCE610
 CWROCE620
 CWROCE630

```

SUBROUTINE MMUL (R,I,M,LL,N,U)
  REAL*4 R(10,10),T(10,10),J(10,10)
  INTEGER M,LL,N
  DO 20 I=1,M
    DO 20 J=1,N
      U(I,J)=C.O
  DO 10 INDEX=1,LL

```

[illegible]

```

C      PHASE TRANSFORMATION
C      Q=-CONJG(A(K,K))/CABS(A(K,K))
C      DO 60 J=K1,NP
C      A(K,J)=C*A(K,J)
C
C      ELIMINATION OF A(K,J),J=K+2,...,N
C      IF (K.EC.N) GO TO 140
C      Z=C.EO
C      DO 80 J=K1,N
C      Z=Z+REAL(A(K,J))*2+AIMAG(A(K,J))*2
C      C(K1)=C.EO
C      IF (Z.LE.TOL) GO TO 130
C      Z=SQRT(Z)
C      C(K1)=Z
C      W=CABS(A(K,K1))
C      G=(1.-C.EO)/W
C      IF (W.NE.C.EO) Q=A(K,K1)/W
C      A(K,K1)=G*(Z+W)
C      DO 110 I=K1,M
C      O=(C.EO,C.EO)
C      DO 90 J=K1,N
C      U=G+CONJG(A(K,J))*A(I,J)
C      Q=G/(Z*(Z+W))
C      DO 100 J=K1,N
C      A(I,J)=A(I,J)-Q*A(K,J)
C      CUNTINCE
C
C      PHASE TRANSFORMATION
C      Q=-CONJG(A(K,K1))/CABS(A(K,K1))
C      DO 120 I=K1,M
C      A(I,K1)=A(I,K1)*Q
C      K=K1
C      GO TO 10
C
C      TOLERANCE FOR NEGLIGIBLE ELEMENTS
C      EPS=0.EO
C      DO 150 K=1,N
C      S(K)=E(K)
C      T(K)=C(K)
C      EPS=AMAX1(EPS,S(K)+T(K))
C      EPS=EPS*ETA
C
C      INITIALIZATION OF U AND V
C      IF (NU.EC.O) GO TO 180
C      DO 170 J=1,NU
C      DO 160 I=1,M
C      U(I,J)=(C.EO,0.EO)

```

```

CWR01100
CWR01110
CWR01120
CWR01130
CWR01140
CWR01150
CWR01160
CWR01170
CWR01180
CWR01190
CWR01200
CWR01210
CWR01220
CWR01230
CWR01240
CWR01250
CWR01260
CWR01270
CWR01280
CWR01290
CWR01300
CWR01310
CWR01320
CWR01330
CWR01340
CWR01350
CWR01360
CWR01370
CWR01380
CWR01390
CWR01400
CWR01410
CWR01420
CWR01430
CWR01440
CWR01450
CWR01460
CWR01470
CWR01480
CWR01490
CWR01500
CWR01510
CWR01520
CWR01530
CWR01540
CWR01550
CWR01560
CWR01570

```

```

170 U(J,J)=(1.EC,0.E0)
180 IF (NV.EQ.0) GO TO 210
DO 200 J=1,NV
DO 190 I=1,N
V(I,J)=(0.EC,0.E0)
200 V(J,J)=(1.EC,0.E0)
C
C QR DIAGONALIZATION
210 DO 380 KK=1,N
K=N1-KK
C
C TEST FOR SPLIT
220 DO 230 LL=1,K
L=K+1-LL
IF (ABS(T(L)).LE.EPS) GO TO 290
IF (ABS(S(L-1)).LE.EPS) GO TO 240
230 CONTINUE
C
C CANCELLATION OF E(L)
240 CS=C.EC
SN=1.E0
LI=L-1
DO 280 I=L,K
F=SN*T(I)
T(I)=CS*T(I)
IF (ABS(F).LE.EPS) GO TO 290
H=S(I)
W=SQRT(F*F+H*H)
S(I)=W
CS=H/W
SN=-F/W
IF (NV.EQ.0) GO TO 260
DO 250 J=1,N
X=REAL(U(J,LI))
Y=REAL(U(J,I))
U(J,LI)=CMPLX(X*CS+Y*SN,0.E0)
250 U(J,I)=CMPLX(Y*CS-X*SN,0.E0)
IF (NP.EQ.N) GO TO 280
DO 270 J=N1,NP
Q=A(LI,J)
R=A(I,J)
A(LI,J)=G*CS+R*SN
270 A(I,J)=R*CS-Q*SN
280 CONTINUE
C
C TEST FOR CONVERGENCE
290 W=S(K)
IF (L.EQ.K) GO TO 360

```

```

CWR01580
CWR01590
CWR01600
CWR01610
CWR01620
CWR01630
CWR01640
CWR01650
CWR01660
CWR01670
CWR01680
CWR01690
CWR01700
CWR01710
CWR01720
CWR01730
CWR01740
CWR01750
CWR01760
CWR01770
CWR01780
CWR01790
CWR01800
CWR01810
CWR01820
CWR01830
CWR01840
CWR01850
CWR01860
CWR01870
CWR01880
CWR01890
CWR01900
CWR01910
CWR01920
CWR01930
CWR01940
CWR01950
CWR01960
CWR01970
CWR01980
CWR01990
CWR02000
CWR02010
CWR02020
CWR02030
CWR02040
CWR02050

```

C

```

C CRIGIN SHIFT
X=S(L)
Y=S(K-1)
G=T(K-1)
H=T(K)
F=((Y-w)*(Y+w)+(G-H)*(G+H))/(2.E0*H*Y)
G=SQR(T(F*F+1.E0)
IF (F.LT.0.E0) G=-G
F=((X-w)*(X+w)+(Y/(F+G)-H)*h)/X

```

C

```

C CR STEP
CS=1.E0
SN=1.E0
LI=L+1 I=LI,K
DO 350
G=T(I)
Y=S(I)
H=SN*G
G=CS*G
W=SQR(T(H*H+F*F)
T(I-1)=W
CS=H/W
SN=H/W+G*SN
F=X*CS-X*SN
G=Y*CS
H=Y*SN
IF (NV.EQ.0) GO TO 310
DO 300
J=I,N
X=REAL(V(J,I-1))
W=REAL(V(J,I))
V(J,I-1)=CMPLX(X*CS+Y*SN,0.E0)
V(J,I)=CMPLX(W*CS-X*SN,0.E0)
W=SQR(T(H*H+F*F)
S(I-1)=W
CS=H/W
SN=H/W+SN*Y
F=CS*G+SN*Y
X=CS*Y-SN*G
IF (NU.EQ.0) GO TO 330
DO 320
J=I,N
Y=REAL(U(J,I-1))
W=REAL(U(J,I))
U(J,I-1)=CMPLX(Y*CS+W*SN,0.E0)
U(J,I)=CMPLX(W*CS-Y*SN,0.E0)
IF (N.EC.NP) GO TO 350
DO 340
J=N1,NP

```

300
310

320
330

CKR02060
 CKR02070
 CKR02080
 CKR02090
 CKR02100
 CKR02110
 CKR02120
 CKR02130
 CKR02140
 CKR02150
 CKR02160
 CKR02170
 CKR02180
 CKR02190
 CKR02200
 CKR02210
 CKR02220
 CKR02230
 CKR02240
 CKR02250
 CKR02260
 CKR02270
 CKR02280
 CKR02290
 CKR02300
 CKR02310
 CKR02320
 CKR02330
 CKR02340
 CKR02350
 CKR02360
 CKR02370
 CKR02380
 CKR02390
 CKR02400
 CKR02410
 CKR02420
 CKR02430
 CKR02440
 CKR02450
 CKR02460
 CKR02470
 CKR02480
 CKR02490
 CKR02500
 CKR02510
 CKR02520
 CKR02530

CWR02540
 CWR02550
 CWR02560
 CWR02570
 CWR02580
 CWR02590
 CWR02600
 CWR02610
 CWR02620
 CWR02630
 CWR02640
 CWR02650
 CWR02660
 CWR02670
 CWR02680
 CWR02690
 CWR02700
 CWR02710
 CWR02720
 CWR02730
 CWR02740
 CWR02750
 CWR02760
 CWR02770
 CWR02780
 CWR02790
 CWR02800
 CWR02810
 CWR02820
 CWR02830
 CWR02840
 CWR02850
 CWR02860
 CWR02870
 CWR02880
 CWR02890
 CWR02900
 CWR02910
 CWR02920
 CWR02930
 CWR02940
 CWR02950
 CWR02960
 CWR02970
 CWR02980
 CWR02990
 CWR03000
 CWR03010

```

      Q=A(I-I,J)
      R=A(I,J)=C*CS+R*SN
      A(I-I,J)=R*CS-Q*SN
      CONTINUE
      T(L)=O.E0
      T(K)=F
      S(K)=X
      GO TO 220

C      CONVERGENCE
360  IF (W.GE.0.E0) GO TO 380
      S(K)=-W
      IF (NV.EG.0) GO TO 380
      DO 370 J=1,N
        V(J,K)=-V(J,K)
      CONTINUE
370
380  C      SCPT SINGULAR VALUES
      DO 450 K=1,N
        S=-1.E0
        J=K
        DO 390 I=K,N
          IF (S(I).LE.G) GO TO 390
          G=S(I)
          J=I
        CONTINUE
390  IF (J.EG.K) GO TO 450
        S(J)=S(K)
        S(K)=G
        IF (NV.EG.0) GO TO 410
        DO 400 I=1,N
          Q=V(I,J)
          V(I,J)=V(I,K)
          V(I,K)=Q
        IF (NU.EG.0) GO TO 430
        DO 420 I=1,N
          G=U(I,J)
          U(I,J)=U(I,K)
          U(I,K)=G
        IF (N.EG.NP) GO TO 450
        DO 440 I=N1,NP
          Q=A(J,I)
          A(J,I)=A(K,I)
          A(K,I)=Q
        CONTINUE
440
450  C      BACK TRANSFORMATION
  
```



```

20      FORMAT (5F12.0)
C      END
C      *****
C      WRITE A REAL MATRIX
C      -----
C      SUBROUTINE WRITE (A,N,K)
C      REAL*4 A(10,10)
C      INTEGER N,K
C      DO 10 I=1,N
C      WRITE (6,20) (A(I,J),J=1,K)
C      CONTINUE
C      RETURN
C      FORMAT (6F12.5)
C      E'D
C      *****
C      CONVERT REAL MATRIX TO COMPLEX FORM
C      -----
C      SUBROUTINE CPLXCV (X,A,N,K)
C      COMPLEX *8X(10,10)
C      REAL*4 A(10,10)
C      INTEGER N,K
C      DO 20 I=1,N
C      DO 10 J=1,K
C      X(I,J)=CMPLX(A(I,J),0.00)
C      CONTINUE
C      RETURN
C      END
C      *****
C      CONVERT REAL PART OF COMPLEX MATRIX TO REAL MATRIX
C      -----
C      SUBROUTINE CPLRFA (X,A,N,K)
C      COMPLEX *8X(10,10)
C      REAL*4 A(10,10)
C      INTEGER N,K
C      DO 20 I=1,N
C      DO 10 J=1,K
C      A(I,J)=REAL(X(I,J))
C      CONTINUE
C      RETURN
C      END
C      *****

```



```

C=====RGAIN (M,NS,NC,NDB,WR,WI,VF,GN,WLL,TCB,W21,LT,C,CI,CT,M
CPT13820
CPT13830
CPT13840
C=====
CPT13850
CPT13860
CPT13870
CPT13880
CPT13890
CPT13900
CPT13910
CPT13920
CPT13930
CPT13940
CPT13950
CPT13960
CPT13970
CPT13980
CPT13990
CPT14000
CPT14010
CPT14020
CPT14030
CPT14040
CPT14050
CPT14060
CPT14070
CPT14080
CPT14090
CPT14100
CPT14110
CPT14120
CPT14130
CPT14140
CPT14150
CPT14160
CPT14170
CPT14180
CPT14190
CPT14200
CPT14210
CPT14220
CPT14230
CPT14240
CPT14250
CPT14260

SUBROUTINE RGAIN (M,NS,NC,NDB,WR,WI,VF,GN,WLL,TCB,W21,LT,C,CI,CT,M
IHS,MT)
C=====
IMPLICIT REAL*4 (A-H,C-Z)
DIMENSION WR(M),WI(M),VF(M,M),GN(NS,NS)
DIMENSION WLL(NS,NS),TCB(M,M),W21(NS,NS),LT(NS),MT(NS)
K=1
KP=1
KN=1
NRZEV=0
NCPZEV=0
IF (K.GT.M) GO TO 210
C-----
C CHECK FOR EIGVAL AT OR NEAR J-OMEGA AXIS TO INCLUDE IN E-L EIGSYS
C TURN FIRST ONE POSITIVE AND SECOND ONE NEGATIVE
C-----
EIGVR=CABS(WR(K))
EIGVR=ABS(WR(K))
IF (EIGVR.GE.1.E-10) GO TO 60
IF (WI(K)) 40,20,40
NRZEV=NRZEV+1
IF (NRZEV.GT.1) GO TO 30
WR(N)=EIGVR
GO TO 20
WR(K)=-EIGVR
WRITE (6,290)
GO TO 150
NCPZEV=NCPZEV+1
IF (NCPZEV.GT.1) GO TO 50
WR(K)=EIGVR
WR(K+1)=EIGVR
GO TO 40
WR(K)=-EIGVR
WR(K+1)=-EIGVR
WRITE (6,300)
GO TO 180
IF (WR(K)) 140,70,70
IF (WI(K)) 110,80,110
CONTINUE
C----- EIGENVECTOR FOR REAL EIGENVALUE, POSITIVE -----
C80 IF (NOB.EQ.0) GO TO 100
C DO 90 J=1,M
C90 TCB(J,KP)=VF(J,K)
C100 KP=KP+1
K=K+1
GO TO 10

```

```

C-----EIGENVECTOR FOR COMPLEX EIGENVALUE, POSITIVE REAL PART-----CPT14270
110 CONTINUE
C110 IF (NCB.EQ.0) GO TO 130
C DO 120 J=1,M
C FR=VF(J,K)
C FI=-VF(J,K+1)
C TCB(J,K,FI)=FR+FI
C120 TCB(J,KP+1)=FR-FI
C130 KP=KP+2
K=K+2
GO TO 10
140 IF (W1(K)) 180,150,180
C-----EIGENVECTOR FOR REAL EIGENVALUE, NEGATIVE REAL PART-----
150 C(KN)=A P(K)
C1(KN)=W1(K)
IF (NCB.NE.0) GO TO 170
KNS=KN+NS
DO 160 J=1,M
TCB(J,KNS)=VF(J,K)
160 KN=KN+1
170 K=K+1
GO TO 10
C-----EIGENVECTOR FOR COMPLEX EIGENVALUE, NEGATIVE REAL PART-----
180 RK=WR(K)
R1=WI(K)
C(KN)=RR
C(KN+1)=RR
C1(KN)=R1
C1(KN+1)=-R1
IF (NCB.NE.0) GO TO 200
KNS=KN+NS
DO 190 J=1,M
FR=VF(J,K)
FI=-VF(J,K+1)
TCB(J,KNS)=FR+FI
190 TCB(J,KNS+1)=FR-FI
200 KN=KN+2
K=K+2
GO TO 10
210 CONTINUE
C-----FORMATION OF W1-----CPT14670
DO 220 I=1,NS
DO 220 J=1,NS
W1(I,J)=TCB(I,J+NS)
220 CT(I,J)=W1(I,J)
C-----FORMATION OF W21-----CPT14730
DO 230 I=1,NS

```

```

230 DO 230 J=1,NS
240 W21(I,J)=TCB(I+NS,J+NS)
240 C240 IF (NDB.EC.O) GO TO 260
C240 DO 250 I=1,NS
C240 DO 250 J=1,NS
C250 W21(I,J)=-TCB(I,J)
250 W11(I,J)=TCB(I+NS,J)
260 C260 CONTINUE
C-----INVERT w11-----
NSC=NS*NS
CALL MINV (NSQ,w11,NS,DETC,LI,MT)
C-----CALCULATE THE RGAIN MATRIX-----
DO 270 IL=1,NS
DO 270 JL=1,NS
GN(IL,JL)=0.FO
DO 270 KL=1,NS
GN(IL,JL)=GN(IL,JL)+W21(IL,KL)*W11(KL,JL)
270 C270 IF (NDB.EC.O) RETURN
C270 DO 280 I=1,NS
C280 DO 280 J=1,NS
C280 CT(I,J)=w11(J,I)
C280 RETURN
C-----
290 FORMAT (IX,51H EULER-LAGRANGE EQUATIONS HAVE A REAL EIGENVALUE AT,
300 114H OK NEAR ZERO.)
300 FORMAT (IX,49H EULER-LAGRANGE EQUATIONS HAVE A COMPLEX PAIR OF ,40
HEIGENVALUES AT OR NEAR THE J-CMEGA AXIS.)
END
C=====
C SUBROUTINE MINV (MSQ,A,N,D,L,M)
C=====
C IMPLICIT REAL*4 (A-H,O-Z)
C DIMENSION A(MSQ),L(N),M(N)
C DOUBLE PRECISION A,D,BIGA,HOLD
C REAL*4 D,BIGA,HOLD
C NM=N*N
C D=1.0D0
C D=1.0EC
C NK=-N
C DO 180 K=1,N
C NK=NK+A
C L(K)=K
C M(K)=K
C KK=NK+K
C BIGA=A(KK)
C DO 20 J=K,N
C IZ=N*(J-1)

```

CPT14740
CPT14750

CPT14760
CPT14770
CPT14780
CPT14790
CPT14800
CPT14810
CPT14820
CPT14830

CPT14840
CPT14850

CPT14860
CPT14870
CPT14880
CPT14890
CPT14900
CPT14910
CPT14920
CPT14930
CPT14940
CPT14950

CPT14960
CPT14970
CPT14980
CPT14990
CPT15000
CPT15010
CPT15020
CPT15030

CPT15040
CPT15050
CPT15060
CPT15070
CPT15080

CPT15090
CPT15100
CPT15110
CPT15120
CPT15130
CPT15140
CPT15150
CPT15160
CPT15170

```

DO 20 I=K,N
IJ=I+I
IF (DABS(BIGA)-DABS(A(IJ))) 10,20,20
IF (ABS(BIGA)-ABS(A(IJ))) 10,20,20
10 BIGA=A(IJ)
L(K)=I
M(K)=J
20 CONTINUE
-----INTERCHANGE ROWS-----
CPT15180
CPT15190
CPT15200
CPT15210
CPT15220
CPT15230
CPT15240
CPT15250
CPT15260
CPT15270
CPT15280
CPT15290
CPT15300
CPT15310
CPT15320
CPT15330
CPT15340
CPT15350
CPT15360
CPT15370
CPT15380
CPT15390
CPT15400
CPT15410
CPT15420
CPT15430
CPT15440
CPT15450
CPT15460
CPT15470
CPT15480
CPT15490
CPT15500
CPT15510
CPT15520
CPT15530
CPT15540
CPT15550
CPT15560
CPT15570
CPT15580
CPT15590
CPT15600
CPT15610
CPT15620
CPT15630

J=L(K)
IF (J-K) 50,50,30
30 KI=K-N
DO 40 I=1,N
KI=KI+N
HOLD=-A(KI)
JI=KI-K+J
A(KI)=A(JI)
40 A(JI)=HOLD
-----INTERCHANGE COLUMNS-----
CPT15350
CPT15360
CPT15370
CPT15380
CPT15390
CPT15400
CPT15410
CPT15420
CPT15430
CPT15440
CPT15450
CPT15460
CPT15470
CPT15480
CPT15490
CPT15500
CPT15510
CPT15520
CPT15530
CPT15540
CPT15550
CPT15560
CPT15570
CPT15580
CPT15590
CPT15600
CPT15610
CPT15620
CPT15630

I=M(K)
IF (I-K) 80,80,60
60 JP=N*(I-I)
DO 70 J=1,N
JK=NK+J
JI=JP+J
HOLD=-A(JK)
A(JK)=A(JI)
70 A(JI)=HOLD
-----DIVIDE COLUMN BY MINUS PIVOT-----
CPT15450
CPT15460
CPT15470
CPT15480
CPT15490
CPT15500
CPT15510
CPT15520
CPT15530
CPT15540
CPT15550
CPT15560
CPT15570
CPT15580
CPT15590
CPT15600
CPT15610
CPT15620
CPT15630

----- (VALUE OF PIVOT ELEMENT IS CONTAINED IN BIGA) -----
80 IF (BIGA) 100,90,100
C90 D=0.0D0
90 D=C.0EC
RETURN

100 DO 120 I=1,N
IF (I-K) 110,120,110
110 IK=NK+I
A(IK)=A(IK)/(-BIGA)
120 CONTINUE
-----REDUCE MATRIX-----
CPT15550
CPT15560
CPT15570
CPT15580
CPT15590
CPT15600
CPT15610
CPT15620
CPT15630

DO 150 I=1,N
IN=NK+I
HOLD=A(IK)
IJ=I-N
DO 150 J=1,N
IJ=IJ+N
IF (I-K) 130,150,130
130 IF (J-K) 140,150,140

```

```

140      KJ=IJ-I+K
150      A(IJ)=HCLD*A(KJ)+A(IJ)
160      CONTINUE
170      KJ=K-N
180      DU 170 J=1,N
190      KJ=KJ+N
200      IF (J-K) 10C,170,160
210      A(KJ)=A(KJ)/BIGA
220      CONTINUE
230      D=D*BIGA
240      A(KK)=(1.0EC)/BIGA
250      CONTINUE
260      K=N
270      K=(K-1) 260,200,200
280      IF (K) 260,200,200
290      I=L(K)
300      IF (I-K) 230,230,210
310      JW=N*(K-1)
320      JP=N*(I-1)
330      DO 240 J=1,N
340      JK=JQ+J
350      HOLD=A(JK)
360      JI=JR+J
370      A(JK)=-A(JI)
380      A(JI)=HCLD
390      J=M(K)
400      IF (J-K) 190,190,240
410      KI=K-N
420      DO 250 I=1,N
430      KI=KI+N
440      HOLD=A(KI)
450      JI=KI-K+J
460      A(KI)=-A(JI)
470      A(JI)=HCLD
480      GU TO 150
490      K=Q
500      RETURN
510      END
520      SUBROUTINE EREXIT (N,A,IERR)
530      EREXIT RETURNS THE NUMBER OF THE EIGENVALUE WHERE HQR2
540      FAILS, THEN STOPS THE PROGRAM.
550      =====
560      =====
570      =====
580      =====
590      =====
600      =====
610      =====
620      =====
630      =====
640      =====
650      =====
660      =====
670      =====
680      =====
690      =====
700      =====
710      =====
720      =====
730      =====
740      =====
750      =====
760      =====
770      =====
780      =====
790      =====
800      =====
810      =====
820      =====
830      =====
840      =====
850      =====
860      =====
870      =====
880      =====
890      =====
900      =====
910      =====
920      =====
930      =====
940      =====
950      =====
960      =====
970      =====
980      =====
990      =====
1000     =====

```

```

CPT3C900
CPT3C910
CPT3C920
CPT3C930
CPT3C940
CPT3C950
CPT3C960
CPT3C970
CPT3C980
CPT3C990
CPT31C00
CPT31C10
CPT31C20
CPT31C30
CPT31C40
CPT31C50
CPT31C60
CPT31C70
CPT31C80
CPT31C90
CPT31100
CPT31110
CPT31120
CPT31130
CPT31140
CPT31150
CPT31160
CPT31170
CPT31180
CPT31190
CPT31200
CPT31210
CPT31220
CPT31230
CPT31240
CPT31250
CPT31260
CPT31270
CPT31280
CPT31290
CPT31300
CPT31310
CPT31320
CPT31330
CPT31340
CPT31350
CPT31360

=====
C SUBROUTINE MATPRT -- DISPLAYS A TWO-DIMENSIONAL ARRAY (16 COLS. MAX) =
C IN VARIABLE SCREEN FORMAT FOR USER EASE IN ROW IDENTIFICATION. =
C=====
C SUBROUTINE MATPRT (PRTT,NROW,NCCL)
C=====
C IMPLICIT REAL*4 (A-H,O-Z)
C DIMENSION PRTT(NROW,NCCL)
C-----
C IF (NCCL.EQ.0) NCCL=1
C IF (NCCL.EQ.1) WRITE(1,1) (5,10) ((PRTT(I,J),J=1,NCCL),I=1,NROW)
C IF (NCCL.EQ.2) WRITE(1,2) (5,20) ((PRTT(I,J),J=1,NCCL),I=1,NROW)
C IF (NCCL.EQ.3) WRITE(1,3) (5,30) ((PRTT(I,J),J=1,NCCL),I=1,NROW)
C IF (NCCL.EQ.4) WRITE(1,4) (5,40) ((PRTT(I,J),J=1,NCCL),I=1,NROW)
C IF (NCCL.EQ.5) WRITE(1,5) (5,50) ((PRTT(I,J),J=1,NCCL),I=1,NROW)
C IF (NCCL.EQ.6) WRITE(1,6) (5,60) ((PRTT(I,J),J=1,NCCL),I=1,NROW)
C IF (NCCL.EQ.7) WRITE(1,7) (5,70) ((PRTT(I,J),J=1,NCCL),I=1,NROW)
C IF (NCCL.EQ.8) WRITE(1,8) (5,80) ((PRTT(I,J),J=1,NCCL),I=1,NROW)
C IF (NCCL.EQ.9) WRITE(1,9) (5,90) ((PRTT(I,J),J=1,NCCL),I=1,NROW)
C IF (NCCL.EQ.10) WRITE(1,10) (5,100) ((PRTT(I,J),J=1,NCCL),I=1,NROW)
C IF (NCCL.EQ.11) WRITE(1,11) (5,110) ((PRTT(I,J),J=1,NCCL),I=1,NROW)
C IF (NCCL.EQ.12) WRITE(1,12) (5,120) ((PRTT(I,J),J=1,NCCL),I=1,NROW)
C IF (NCCL.EQ.13) WRITE(1,13) (5,130) ((PRTT(I,J),J=1,NCCL),I=1,NROW)
C IF (NCCL.EQ.14) WRITE(1,14) (5,140) ((PRTT(I,J),J=1,NCCL),I=1,NROW)
C IF (NCCL.EQ.15) WRITE(1,15) (5,150) ((PRTT(I,J),J=1,NCCL),I=1,NROW)
C IF (NCCL.EQ.16) WRITE(1,16) (5,160) ((PRTT(I,J),J=1,NCCL),I=1,NROW)
C RETURN
C-----
C 10 FORMAT (F12.5)
C 20 FORMAT (2F12.5)
C 30 FORMAT (3F12.5)
C 40 FORMAT (4F12.5)
C 50 FORMAT (5F12.5)
C 60 FORMAT (6F12.5)
C 70 FORMAT (6F12.5,/,F12.5,/)
C 80 FORMAT (6F12.5,/,2F12.5,/)
C 90 FORMAT (6F12.5,/,3F12.5,/)
C 100 FORMAT (6F12.5,/,4F12.5,/)
C 110 FORMAT (6F12.5,/,5F12.5,/)
C 120 FORMAT (6F12.5,/,6F12.5,/)

```

```

100 FORMAT (6F12.5,/,6F12.5,/,F12.5,/,/)
1100 FORMAT (6F12.5,/,6F12.5,/,2F12.5,/,/)
1200 FORMAT (6F12.5,/,6F12.5,/,3F12.5,/,/)
1300 FORMAT (6F12.5,/,6F12.5,/,4F12.5,/,/)
END
C=====
C SUBROUTINE MODE (WNORM,G,GNORM,NS,N1,N2,ICON)
C
C   WNORM TRANSFORMATION MATRIX U OR U-INV
C   NS NO. OF STATE
C   NC NO. OF INPUTS OR OUTPUTS
C   ICON CONTROL FLAG TO INDICATE WHICH TRANSFORMATION
C      0 = MODAL G
C      1 = MODAL GAMMA
C      2 = MODAL H
C      3 = MODAL C
C      4 = MODAL K
C      5 = CONTROL EIGENVECTOR MATRIX
C      6 = MEASUREMENT EIGENVECTOR MATRIX
C=====
C IMPLICIT REAL*4(A-F,G-Z)
C DIMENSION WNORM(NS,NS),G(N1,N2),GNORM(N1,N2)
C DO 10 I=1,N1
C DO 10 J=1,N2
C   GNORM(I,J)=C.
C   IPCINT=IPCINT+1
C GO TO (20,20,90,90,20,90,90), IPCINT
C DO 20 J=1,N2
C DO 30 I=1,NS
C DO 30 K=1,NS
C   GNORM(I,J)=GNORM(I,J)+WNORM(I,K)*G(K,J)
C GO TO (40,70,90,90,80), IPCINT
C WRITE (6,170)
C DO 60 I=1,NS
C WRITE (6,230) (GNORM(I,J),J=1,N2)
C RETURN
C WRITE (6,180)
C GO TO 50
C WRITE (6,240)
C GO TO 50
C DO 100 J=1,NS
C DO 100 I=1,N1
C DO 100 K=1,NS
C   GNORM(I,J)=GNORM(I,J)+G(I,K)*WNORM(K,J)
C GO TO (110,110,110,120,110,130,140), IPCINT
C WRITE (6,190)
C GO TO 150
C WRITE (6,200)
CPT131370
CPT131380
CPT131390
CPT131400
CPT131410
CPT131420
CPT131430
CPT131440
CPT131450
CPT131460
CPT131470
CPT131480
CPT131490
CPT131500
CPT131510
CPT131520
CPT131530
CPT131540
CPT131550
CPT131560
CPT131570
CPT131580
CPT131590
CPT131600
CPT131610
CPT131620
CPT131630
CPT131640
CPT131650
CPT131660
CPT131670
CPT131680
CPT131690
CPT131700
CPT131710
CPT131720
CPT131730
CPT131740
CPT131750
CPT131760
CPT131770
CPT131780
CPT131790
CPT131800
CPT131810
CPT131820
CPT131830
CPT131840
CPT131850
CPT131860
CPT131870
CPT131880
CPT131890
CPT131900
CPT131910
CPT131920
CPT131930
CPT131940
CPT131950
CPT131960
CPT131970
CPT131980
CPT131990
CPT132000
CPT132010
CPT132020
CPT132030
CPT132040
CPT132050
CPT132060
CPT132070
CPT132080
CPT132090
CPT132100
CPT132110
CPT132120
CPT132130
CPT132140
CPT132150
CPT132160
CPT132170
CPT132180
CPT132190
CPT132200
CPT132210
CPT132220

```



```

C
DO 50 K=1,NS
IF (KK.EC.1) GO TO 40
IF (DABS(WY(K)).LT.1.D-10) GO TO 50
IF (ABS(WY(K)).LT.1.E-10) GO TO 50
LC=LC+1
EMAX=0.50
DO 20 I=1,NS
CMCD=VEC(I,K)**2+VEC(I,K+1)**2
IF (CMCD-EMAX) 20,10,10
EMAX=CMCD
M=I
CONTINUE
VMR=VEC(M,K)
VM1=VEC(V,K+1)
DO 30 I=1,NS
VR=VEC(I,K)
VI=VEC(I,K+1)
VECRN=(VR*VMR+VI*VM1)/EMAX
VEGIN=(-VR*VM1+VI*VMR)/EMAX
WIGRM(I,K)=VECRN
WIGRM(I,K+1)=VEGIN
CONTINUE
KK=1
GO TO 50
40
50
C-----NORMALIZE REAL EIGENVECTORS BY THE TOTAL LENGTH-----
C
DO 80 K=1,NS
IF (DABS(WY(K)).GE.1.D-10) GO TO 80
IF (ABS(WY(K)).GE.1.E-10) GO TO 80
LR=LR+1
REMOD=C.E0
DO 60 I=1,NS
REMOD=VEC(I,K)**2+REMOD
RMCD=DSQRT(REMOD)
RMCD=SQRT(REMOD)
DO 70 I=1,NS
RVEC=VEC(I,K)/RMCD
WIGRM(I,K)=RVEC
CONTINUE
GL TO (50,100,110,120,130), IWRITE
WR ITE (6,320)
GO TO 140
WR ITE (6,330)
GO TO 140
WR ITE (6,340)
GO TO 140
10
20
30
40
50
C-----
C
60
C
70
80
90
100
110
120
130
140
150
CPT17710
CPT17720
CPT17730
CPT17730
CPT17740
CPT17750
CPT17760
CPT17770
CPT17780
CPT17790
CPT17800
CPT17810
CPT17820
CPT17830
CPT17840
CPT17850
CPT17860
CPT17870
CPT17880
CPT17890
CPT17900
CPT17910
CPT17920
CPT17930
CPT17940
CPT17950
CPT17960
CPT17970
CPT17980
CPT17990
CPT18000
CPT18010
CPT18020
CPT18030
CPT18040
CPT18050
CPT18060
CPT18070
CPT18080
CPT18090
CPT18100
CPT18110
CPT18120
CPT18130
CPT18140
CPT18150

```

```

120 WRITE (6,350)
130 GO TO 140
140 WRITE (6,360)
      KK=0
      NPRTW=C
      NPRTW=1
      DO 180 I=1,NS
      IF (KK.EC.1) GO TO 170
      IF (DABS(WY(I)).GT.1.E-10) KK=1
      IF (DABS(WY(I)).GT.1.E-10) KK=1
      C-----PRINT OUT NO MORE THAN 6 WORDS, NOT SEPARATING COMPLEX EIGVAL-----
      IF (NPRTW.LT.5.OR. (NPRTW.EQ.5.AND.KK.EQ.0)) GO TO 150
      FMT(NPRTW+1)=RIGHT
      WRITE (6,FMT) (STOPE(J),J=1,NPRTW)
      NPRTW=C
      NPRTW=1
      NPRTW=NPRTW+1
      NPRTW=NPRTW+1
      IF (KK.EC.1) GO TO 160
      STOPE(NPRTW)=WZ(I)
      FMT(NPRTW)=FIELD
      NPRTW=NPRTW+1
      FMT(NPRTW)=SEMCOL
      GO TO 180
      STOPE(NPRTW)=WZ(I)
      FMT(NPRTW)=FIELD
      FMT(NPRTW+1)=COMMA
      STOPE(NPRTW+1)=WY(I)
      FMT(NPRTW+2)=FIELD
      FMT(NPRTW+3)=SEMCOL
      NPRTW=NPRTW+3
      NPRTW=NPRTW+1
      GO TO 180
      KK=0
      CONTINUE
      FMT(NPRTW)=SEMCOL
      FMT(NPRTW+1)=RIGHT
      WRITE (6,FMT) (STOPE(J),J=1,NPRTW)
      IF (IWRITE.NE.1) GO TO 190
      WRITE (6,370)
      GO TO 200
      CONTINUE
      WRITE (6,380)
      CONTINUE
      CALL RAPRINT (NS,NS,NS,6,WNGRM,4, (6(1X,1PE13.6)))
      GO TO (230,210,220,220), IWRITE
      CALL MCODE (KNORM,H0,CM,NS,N1,N2,5)
      GO TO 230
190
200
210

```


WURZCC12
 WURZCC13
 WURZCC14
 WURZCC15
 WURZCC16
 WURZCC17
 WURZCC18
 WURZCC19
 WURZCC20
 WURZCC21
 WURZCC22
 WURZCC23
 WURZCC24
 WURZCC25
 WURZCC26
 WURZCC27
 WURZCC28
 WURZCC29
 WURZCC30
 WURZCC31
 WURZCC32
 WURZCC33
 WURZCC34
 WURZCC35
 WURZCC36
 WURZCC37
 WURZCC38
 WURZCC39
 WURZCC40
 WURZCC41
 WURZCC42
 WURZCC43
 WURZCC44
 WURZCC45
 WURZCC46
 WURZCC47
 WURZCC48
 WURZCC49
 WURZCC50
 WURZCC51
 WURZCC52
 WURZCC53
 WURZCC54
 WURZCC55
 WURZCC56
 WURZCC57
 WURZCC58
 WURZCC59

CALL WURZEL(A,B,N,THETA,EPS,ND)
 DESCRIPTION OF PARAMETERS
 A - SYMMETRIC, POSITIVE-DEFINITE, INPUT SQUARE MATRIX.
 B - SYMMETRIC, CONTAINS SQUARE ROOT OF A, AFTER EXIT
 FROM WURZEL.
 N - ORDER OF A AND B.
 THETA - CONSTANT (SEE METHOD AND REMARKS). IF EIGENVALUES
 OF A ARE UNKNOWN, CHOOSE THETA CLOSE TO UNITY.
 EPS - CONSTANT SPECIFYING DESIRED ACCURACY (SEE METHOD).
 ND - FOR SIX PLACE ACCURACY CHOOSE EPS = 1.E-6
 - ORDER OF A AND B.

REMARKS
 PARAMETERS A, B, THETA, AND EPS ABOVE ARE ALL REAL*8
 THE CONVERSION PROCESS IS VERY SLOW IF THE LARGEST AND
 SMALLEST EIGENVALUES OF A DIFFER BY SEVERAL ORDERS OF
 MAGNITUDE.

SUBROUTINES AND FUNCTION SUBPROGRAM REQUIRED
 NCNE

METHOD
 IF X IS THE VECTOR ASSOCIATED WITH MATRIX A, THEN MATRIX A
 IS POSITIVE-DEFINITE IF X^*AX IS GREATER THAN ZERO.
 NEW THE SEQUENCE:

$$B(K+1) = B(K) + C*(A - B(K)**2) \quad \dots 1..$$
 CONVERGES TO THE SQUARE ROOT OF A.
 IN EXPRESSION 1.. $C = \text{THETA}/2 * \text{SQRT}(\text{NORM}(A))$, WHERE:
 THETA IS A CONSTANT GREATER THAN ZERO AND LESS THAN THE
 SQUARE ROOT OF ALPHA.
 ALPHA IS THE QUOTIENT OF NORM(A), DIVIDED BY THE MAXIMUM
 EIGENVALUES OF A, AND IT IS GREATER OR EQUAL TO ONE.
 NORM(A) IS THE MAXIMUM VALUE OBTAINED BY ADDING EACH ROW
 OF A ALL THE WAY ACROSS.
 THE FIRST APPROXIMATION OF B, THE SQUARE ROOT MATRIX, IS
 OBTAINED AS FOLLOWS; $B(0) = 2.C * C * A$
 THE RATE OF CONVERGENCE OF 1.. TO THE SQUARE ROOT OF A
 IS GIVEN BY THE RATE OF CONVERGENCE TO ZERO OF THE
 SEQUENCE:

$$X(K) = (1 - \text{THETA}(\text{SQRT}(\text{LAMDAMIN}/\text{LAMDAMAX})))**K$$
 WHERE LAMDAMIN AND LAMDAMAX ARE THE MINIMUM AND MAXIMUM
 EIGENVALUES OF A, RESPECTIVELY.
 TC ACCELERATE CONVERGENCE THETA SHOULD BE CHOSEN CLOSE TO
 THE SQUARE ROOT OF ALPHA

WURZC107
 WURZC108
 WURZC109
 WURZC110
 WURZC111
 WURZC112
 WURZC113
 WURZC114
 WURZC115
 WURZC116
 WURZC117
 WURZC118
 WURZC119
 WURZC120
 WURZC121
 WURZC122
 WURZC123
 WURZC124
 WURZC125
 WURZC126
 WURZC127
 WURZC128
 WURZC129

C COMPUTE MAXIMUM VALUE AS SHOWN IN PREVIOUS COMMENT.

DO 75 J=I,N
 S=ABS(E(I,J)-BB(J))
 IF (S-DELTA) 75,75,74

74 DELTA=S

75 WRITE(6,1000)DELTA

80 B(I,J)=BB(J)
 CONTINUE

SET SYMMETRIC TERMS IN MATRIX B.

NN=N-1
 DO 90 I=1,NN

NJ=I+1

DO 90 J=NJ,N

90 B(J,I)=E(I,J)

WRITE(6,1001)((B(I,LL),LL=1,N),II=1,N)
 IF (DELTA-EPS) 95,95,94

94 GO TO 50

95 RETURN

END

C *****
 C ***** THE END *****
 C *****
 C *****

LIST OF REFERENCES

1. Kalman, R. E., "When is a Linear Control System Optimal?," J. of Basic Eng, Tran. ASME, Vol 86, pp. 51-60, March 1964.
2. Athans, M., "The Role and Use of the Stochastic Linear Quadratic Gaussian Problem in Control System Design", IEEE Tran. Automat. Control, vol AC-16, pp. 529-552, December 1971.
3. Bryson, A. E. and Y. Ho., Applied Optimal Control, Waltham, MA, Ginn, 1969.
4. Kwakernaak, H., and Sivan, R., Linear Optimal Control System, Wiley International, New York, 1972.
5. Tyler, J. S., "The Use of a Quadratic Performance Index to Design Multivariable Control System," IEEE Trans. Auto. Control, Vol 11, pp. 84-92, 1966
6. Fallside, F. and Seraji, H., "Design of Optimal Systems by a Frequency Domain Technique," Proc. IEE, Vol 117, pp. 2017-2024, 1970.
7. MacFarlane, A. G. J., "Return Difference and Return Ratio Matrices and Their Use in Analysis and Design of Multivariable Feedback System," Proc. IEE, Vol 117, pp. 2037-2049, 1970.
8. Solheim, O. A., "Design of Optimal Control Systems with Prescribed Eigenvalues," Int. J. Controls, Vol. 15, No. 1, pp. 143-160, 1972.
9. Harvey, C. A. and Stein, G., "Quadratic Weights for Asymptotic Regulator Properties," IEEE tran. Auto. Control, Vol. AC-23, June 1978.
10. Stein, G., "Generalized Quadratic Weights for Asymptotic Regulator Properties," IEEE Tran. Auto. Control, Vol AC-24, pp. 559-567, August 1979.
11. Broussard, J. R., "A Quadratic Weight Selection Algorithm," Proc. of the 1981 Joint Auto. Contr. Conference, 1981.
12. Gully, S. W., Sandell, N. R., and Lee, W. H., "A Robustness Synthesis Method," IEEE Decision and Control Proc. 21, 1982.

13. Dolye, J. C., and Stein, G., "Multivariable Feedback Design: Concepts for a Classical/Modern Synthesis," IEEE Trans. Auto. Control, Vol. 26, No. 1 PP. 4-16, Feb 1981.
14. Safanov, M. G. and Athans, M., "Gain and Phase Margin of Multiloop LQG Regulars," IEEE Trans. Auto. Control, April 1977.
15. Lehtomaki, N.A., N.R. Sandell, Jr. and M. Athans, "Robustness Results in Linear Quadratic Gaussian Based Multivariable Control Designs," IEEE Trans. Auto. Contr., Vol. AC-26, No.1, Feb. 1981
16. Moore, B. C., "On the Flexibility Offered by State Feedback in Multivariable Systems Beyond Closed Loop Eigenvalue Assignment," IEEE Tran. Auto. Contr., AC-20, pp. 689-692, October 1976.
17. Dooren, P. V., Kautsky, J. and Nichols, N. K., "Robust Pole Assignment in Linear State Feedback," Int. J Contr. Vol 41. No 5. pp. 1129-1155, 1985.
18. Kwakernaak, H., Asymptotic Root Loci for Multivariable Linear Optimal Regulator, IEEE Tran. Auto. Contr., AC-21, pp. 378-381 1975.
19. NASA TM-84524, Application of Matrix Singular Value Properties for Evaluating Gain and Phase Margins of Multiloop Systems, Mukhopadhyay, V. and Newsom, J. R., July 1982.
20. Vanderplatts, G. N., ADS - A Fortran Program for Automated Design Synthesis, - Version 1.0, Naval Postgraduate School, Monterey, California, May 1984.
21. Vanderplatts, G. N., Numerical Optimization Techniques for Engineering Design: With Applications, McGraw Hill, 1984.
22. Gordon, V. C., Utilization of Numerical Optimization Techniques in the Design of Robust Multi-input Multi-output Control Systems, Ph.D. Dissertation, Naval Postgraduate School, Monterey, California 1984.
23. Alpatech Inc., TR-121, Multivariable Stability Margins for Vehicle Flight Control System, Sandell, N. R., Jr., et al, Dec. 1981
24. NASA Report 3889, Design of Multivariable Feedback Control Systems Via Spectral Assignment Using Reduced-Order Models and Reduced-Order Observers, R. R. Mielke, L. J. Tung and P. I. Carraway III, April 1985

INITIAL DISTRIBUTION LIST

	No.	Copies
1. Defense Technical Information Center Cameron Station Alexandria, Virginia 22304-6145		2
2. Library, Code 0142 Naval Postgraduate School Monterey, California 93943-5100		2
3. Department Chairman, Code 67 Department of Aeronautics Naval Postgraduate School Monterey, California 93943-5100		1
4. Prof. D. J. Collins, Code 67Co Department of Aeronautics Naval Postgraduate School Monterey, California 93943-5100		2
5. Mr. Chow Wah Keh #08-5067, Blk 639, Ang Mo Kio Ave. 6 S'pore 2056 Republic of Singapore		3

216193

Thesis
C448 C448878
c.1 c.1

Chow

Automated pole
placement algorithm
for multivariable
optimal control
synthesis.

8 NOV 68

32404

216193

Thesis
C448878
c.1

Chow

Automated pole
placement algorithm
for multivariable
optimal control
synthesis.



thesC448878

Automated pole placement algorithm for m



3 2768 000 64784 6

DUDLEY KNOX LIBRARY